

UNIVERZITA KARLOVA V PRAZE

Přírodovědecká fakulta

Katedra aplikované geoinformatiky a kartografie



**NÁVRH WEBOVÉHO PORTÁLU AKTUÁLNÍ
METEOROLOGICKÉ SITUACE S VYUŽITÍM DRUŽICOVÝCH
DAT**

**THE DESIGN OF AN UP-TO-DATE METEOROLOGIC
SITUATION WEB PORTAL WITH THE USE OF SATELLITE
IMAGES**

Diplomová práce

Lucie Hrstková

srpen 2009

Vedoucí diplomové práce: Mgr. Stanislav Grill

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně a že jsem všechny použité prameny řádně citovala.

Jsem si vědoma toho, že případné použití výsledků, získaných v této práci, mimo Univerzitu Karlovu v Praze je možné pouze po písemném souhlasu této univerzity.

Svoluji k zapůjčení této práce pro studijní účely a souhlasím s tím, aby byla řádně vedena evidenci vypůjčovateli.

V Praze dne 30. srpna 2009

.....
Lucie Hrstková

Poděkování

Na tomto místě bych ráda poděkovala vedoucímu mé práce Mgr. Stanislavu Grillovi za věnovaný čas, cenné rady a připomínky. Dále pak bych chtěla poděkovat Mgr. Stanislavu Rackovi za poskytnutí informací. V neposlední řadě děkuji rodině za podporu v průběhu celého studia.

Návrh webového portálu aktuální meteorologické situace s využitím družicových dat

Abstrakt

Cílem diplomové práce je vytvoření webového portálu s prezentací aktuálních snímků Meteosatu a doprovodných aplikací. Portál je určen pro „běžného“ uživatele. Součástí portálu bude i galerie (archiv) zajímavých událostí z pohledu meteorologie (bouřky, různý druh oblačnosti,), zároveň bude portál obsahovat popis různých meteorologických situací a objektů ve snímku. Kromě prezentace snímků zde bude uživatelům k dispozici i databáze s předchozími snímky s možností zobrazit si ty zvolené, odkazy na jiné stránky s podobnou problematikou, informace o družici, ze které jsou snímky pořízeny, možnost zobrazit snímky se zajímavými nebo výjimečnými ději. Prezentace bude doplněna slovním komentářem aktuální situace.

The design of an up-to-date meteorologic situation web portal with the use of satellite images

Abstract

The aim of this master thesis is to create a web portal with the presentation of up-to-date Meteosat images and accompanying applications. The web portal is intended for a „common“ user. It will contain a registry of events which are interesting in terms of meteorology (storms, various types of clouds, etc.) as well as descriptions of different meteorologic situations and objects in images. Except for the presentation of the images, this web portal will provide its users with a database of previous images, which can be displayed, links to other websites with a similar topic, informations about the satellite where these images were taken from and the opportunity to display the images of interesting or rare processes. A commentary on the up-to-date situation will complement the presentation.

Obsah

1 ÚVOD, CÍLE, HYPOTÉZY	7
1.1 Úvod.....	7
1.2 Cíle práce	8
1.2.1 Hlavní cíl.....	8
1.2.2 Vedlejší cíle.....	8
1.2.3 Hypotézy	8
2 REŠERŠE	9
2.1 Literatura zabývající se databázemi	9
2.2 Literatura zabývající se RIA technologiemi.....	9
2.3 Literatura meteorologická	10
2.4 Webové stránky s animací snímků MSG	11
2.5 Flexové aplikace pro GIS.....	12
3 ÚVOD DO PROBLEMATIKY.....	14
3.1 Databáze.....	14
3.1.1 Základní pojmy	14
3.1.2 Historie databází.....	15
3.1.3 Relační databáze.....	16
3.2 Rich Internet Applications - RIA	17
3.3 Meteorologie	19
3.3.1 Základní pojmy	19
3.3.2 Historie pozorování a klasifikace oblaků.....	20
3.3.3 Meteosat Second Generation – MSG	21
3.3.4 Distribuce meteorologických dat	22
4 PRINCIP TVORBY WEBOVÉHO PORTÁLU AKTUÁLNÍ METEOROLOGICKÉ SITUACE.....	23
4.1 MySQL.....	24
4.2 Návrh relační databáze.....	24
4.1.1 Plánování.....	25
4.1.2 Sběr požadavků	25
4.1.3 Konceptuální návrh	26

4.1.4 Logický návrh	27
4.1.5 Fyzický návrh.....	28
4.2 Adobe Flex	28
4.2.1 Adobe Flex Builder 3	29
4.2.2 Vytvoření projektu v Adobe Flex.....	31
4.2.3 MXML a actionscript	32
4.2.5 Flex a php.....	33
4.3 Práce se snímky MSG	34
4.3.1 Zpracování dat.....	34
4.3.2 Klasifikace oblačnosti	34
5 POSTUP TVORBY WEBOVÉHO PORTÁLU AKTUÁLNÍ METEOROLOGICKÉ SITUACE.....	38
5.1 Vytvoření databáze	38
5.2 Připojení databáze k webu	40
5.2.1 PHP	41
5.3 Webový portál.....	42
5.3.1 Vzhled portálu.....	42
5.3.2 Animace snímků.....	44
5.4 Klasifikace oblačnosti	46
6 DISKUZE.....	49
7 ZÁVĚR.....	51
SEZNAM POUŽITÝCH ZKRATEK	52
SEZNAM OBRÁZKŮ, TABULEK A PŘÍLOH	53
Seznam obrázků	53
Seznam tabulek	53
Seznam příloh	54
SEZNAM ZDROJŮ INFORMACÍ	55

1 ÚVOD, CÍLE, HYPOTÉZY

1.1 Úvod

Geoinformatika je vědním oborem, jehož cílem je sběr prostorových dat, jejich zpracování a následná prezentace, na kterou bude zaměřena i tato diplomová práce. Publikování dat v prostředí internetu je dnes považováno za samozřejmost a díky tomu také jedna z nejvíce diskutovaných oblastí v mnoha odvětvích. Ani sebelepší data by nám neposkytla potřebné informace, kdybychom je nedokázali vhodně prezentovat. A protože internet je v současné době nejrozsáhlejším a nejpopulárnějším médiem a pro jeho využití stačí základní práce s počítačem, je právě on nejvhodnějším médiem pro publikaci dat, tedy i geodat.

V současné době se stále častěji setkáváme s pojmem RIA – Rich Internet Application, překládaným jako chytré či bohaté internetové aplikace. Poskytují možnosti, které jdou daleko za pouhé čtení. Díky RIA lze mnohem efektivněji ovládat uživatelské rozhraní a zlepšit tak komfort pro uživatele. Je jen otázkou času, kdy si RIA technologie najdou své uplatnění v geoinformatice. První náznaky se již objevily. Dalším bude tato práce, která by měla prezentovat data DPZ uživatelům pomocí RIA technologie.

Úvod této práce se zaměřuje na teoretický pohled na danou problematiku. Pozornost je věnována všem vědním disciplínám, do kterých práce zasahuje, tedy databázím, RIA technologiím a meteorologii.

V další části práce bude proveden návrh ukládání a manipulace se snímky. Vytvořená databáze snímků bude specifická pro potřeby informačního portálu. Posléze půjde o návrh automatické tvorby animace ze snímků, která bude základem webového portálu navrženého v prostředí Adobe Flex. Dále budou do portálu začleněny i doprovodné funkce. Společně s tím bude přihlédnuto i k možnosti integrace GIS vrstev v rámci aplikace. V neposlední řadě bude navržen způsob automatické klasifikace oblačnosti na snímku.

V závěru práce bude provedena diskuze, v níž budou zhodnoceny dosažené výsledky a porovnání s již existujícími aplikacemi, nastínění problémů a možných vylepšení, které přesahují rámec práce.

Práce by měla být přínosem těm, kdož chtějí efektivně prezentovat svá data na internetu a nespokojí se s dosud používanými metodami. Samotná aplikace je pak určena pro „běžného“

uživatele internetu, kterému by měla ve vstřícné formě zprostředkovat data DPZ, jejich přínos a být i impulsem možného hlubšího zájmu.

1.2 Cíle práce

1.2.1 Hlavní cíl

Vytvoření webového portálu s prezentací aktuálních snímků Meteosatu a doprovodných aplikací s použitím RIA technologie.

Vstupem pro portál budou data z vytvořené databáze družicových snímků. V prostředí Adobe Flex pak bude navržen vzhled portálu a naprogramována samotná prezentace i další funkce portálu.

1.2.2 Vedlejší cíle

1) Návrh databáze snímků

- Zvolit vhodnou databázi pro práci se snímky (kapitola 4.1).
- Vybrat vhodné snímky.

2) Automatická tvorba animací ze snímků

- Práce s actionscriptem (kapitola 5.3.2).

3) Návrh vzhledu portálu

- Práce v Adobe Flex (kapitola 5.3.1).
- Vytvořit doprovodné aplikace.

4) Klasifikace oblačnosti ze snímků

- Vytvoření algoritmu pro zjištění pokrytí oblačností (kapitola 5.4).
- Návrh komentáře aktuální meteorologické situace (hodnocení pokrytí oblačností).

1.2.3 Hypotézy

- 1) Zvolený produkt a jeho automatická klasifikace budou přehledně vypovídat o stavu oblačnosti
- 2) Vybrané prostředí (Adobe Flex) je vhodné pro tvorbu informačního portálu s využitím družicových dat

2 REŠERŠE

2.1 Literatura zabývající se databázemi

OPPEL, A. (2006):

Databáze bez předchozích znalostí. 1. Vydání, Computer Press, Brno, 320 str.

Tato kniha oslovuje jak začátečníky, kteří se dosud s databázovými technologiemi nesetkali, tak i jejich zkušenější kolegy. Autor ve své knize staví na svých vlastních obsáhlých zkušenostech s návrhem, administrací i školením databází. Díky přívětivému výkladu pomáhá porozumět databázím a pochopit veškeré principy jejich návrhu, modelování, fungování, vytváření a používání v běžných aplikacích. Autor popisuje, jak vytvořit efektivní návrh databázové struktury a jak převést realitu do tabulek a vztahů.

V jednotlivých kapitolách autor nejprve hovoří o základních pojmech a definicích ze světa databází, rozebírá konceptuální, logické a fyzické komponenty relačního modelu a před samotným návrhem databáze nás ještě seznamuje s jazykem SQL. V nejdůležitější části knihy, využitě v této diplomové práci, se autor zabývá logickým návrhem databáze a její normalizací, datovým a procesním modelováním a v posledním kroku fyzickým návrhem databáze. Na závěr se zmiňuje o bezpečnosti databází a o databázích pro online analytické zpracování.

Součástí knihy jsou i příklady z databázového systému Microsoft Access a Oracle. Každou kapitulu zakončuje také test znalostí.

2.2 Literatura zabývající se RIA technologiemi

NOBLE, J.; ANDERSON, T. (2008):

Flex 3 Cookbook: Code Recipes, Tips, and Tricks for RIA Developers. 1. Vydání, Adobe Developer Library, 704 str.

Tato velice praktická kniha obsahuje více než 300 osvědčených návodů na interaktivní RIA aplikace. Najdete zde všechno od základů Flexu, práce s menu a ovládacími prvky, přes metody

pro kompilování, rozvinutí a konfiguraci Flex aplikací. U každého návodu je vysvětlena funkčnost a použití a spoustu z nich nabízí i zdrojové kódy.

V jednotlivých kapitolách se autoři věnují menu a ovládacím prvkům, práci s textem, datové mřížce, používání obrázků, videa a zvuků, CSS, práci se soubory, poli, a správou dat, validací a formátováním, užívání grafů a vizualizací dat, obsluhou a přístupem k datům a nakonec i práci s Adobe AIR.

Tato kniha bude průvodcem po většinu doby této práce. Jelikož aplikací vytvořených ve Flexu zatím není mnoho, bude převážná část informací při tvorbě portálu čerpána z této publikace.

HERRINGTON, J.; KIM, E. (2008):

Getting Started with Flex 3. 1. vydání, Adobe Developer Library, 143 str.

Tato kniha je další zdrojem informací pro tvorbu portálu. Srozumitelným výkladem krok po kroku objasňuje webové služby, správu událostí, návrh vlastního rozhraní a řadu dalších. Jsou zde ukázky některých RIA projektů vytvořených Flexem, popis práce s ActionScriptem a MXML, spolupráce Flexu s ASP.NET, PHP a J2EE, návrhy na vlastní aplikace, jako vytváření 3D grafů, mapových aplikací a jiných.

WARD, J.; COENRAETS, CH.; WILSON, G. (2009):

Tour de Flex. Platform Evangelist team

Tour de Flex je AIR aplikace, která obsahuje více než 200 vzorových mini aplikací včetně zdrojového kódu. Celý obsah je rozdělen do několika kategorií: Flex Core Components, Flex Data Access, AIR Capabilities, Cloud APIs, Data Visualization, Mapping a Others components. Tuto aplikaci lze doporučit začátečníkům, ale i pokročilejším uživatelům, kteří v ní jistě najdou inspiraci pro další práci.

2.3 Literatura meteorologická

BEDNÁŘ, J. (2003):

Meteorologie. 1. vydání, Portál, Praha, 224 str.

Tato kniha přináší základní poznatky o zemské atmosféře, dějích v ní probíhajících a utvářejících jak počasí, tak klima. Autor postupně probírá složení a členění atmosféry Země a základní meteorologické a klimatologické prvky, jako jsou tlak a proudění vzduchu,

atmosférické fronty, sluneční záření, radiační a tepelná bilance, teplotní inverze v atmosféře, oblaky a srážky. Seznamuje čtenáře s tematikou týkající se předpovědi počasí, znečištění ovzduší, ozonu v atmosféře, klasifikace a modelování klimatu aj. Výklad je doplněn řadou příkladů a obrázků.

DVOŘÁK, P. (2001):

Atlas oblaků. 1. vydání, Svět křidel, Cheb, 128 str.

Atlas oblaků je stěžejním dílem pro tu část práce, která se zabývá komentářem aktuální meteorologické situace. Hlavním cílem této publikace je popis oblaků, jejich složení, výskytu a vlivu na počasí. Částečně se dotýká i s vyhodnocení prognózy počasí podle oblaků.

V první části jsou popsány hlavní typy oblaků, rozříděné podle mezinárodní klasifikace do jednotlivých skupin a druhů. Druhá část knihy je částí obrazovou. Je však nasnadě, že shromáždění všech obrazových podkladů bylo poměrně zdlouhavou a podrobnou prací, neboť některé druhy oblaků se objevují jen v určité sezóně nebo v konkrétním prostředí.

Atlas je vítanou pomůckou všem meteorologům a meteorologickým pozorovatelům, přírodovědcům, pilotům, leteckým sportovcům a všem zájemcům o krásné útvary, kterými oblaka bezesporu jsou.

2.4 Webové stránky s animací snímků MSG

ČHMÚ

www.chmi.cz/meteo/sat/

Oficiální stránky družicového oddělení Českého hydrometeorologického ústavu. K nalezení jsou zde informace i aktuální snímky. Animace aktuálních snímků MSG, zobrazujících z finančních důvodů pouze Českou republiku a její nejbližší okolí, je řešena pomocí DHTML a javascriptu. Uživatel má na výběr čtyři produkty. Klasický černobílý IR snímek ve formátu střední Evropy a tři produkty ve formátu Česká republika – IR snímek se škálou teplot nejvyšší oblačnosti (IR BT), barevnou kombinaci snímků blízkou reálným barvám (VIS-IR), a tzv. 24-hodinový mikrofyzikální produkt zobrazující různé typy oblačnosti odlišnou barvou (24h-MF). Uživatel si může zvolit, které snímky chce zobrazit, maximálně však dvanáct snímků. Na výběr má snímky posledních třech dní. Nad animací si dále může nechat zobrazit státní hranice, rovnoběžky a poledníky a česká letiště s mezinárodním provozem. Samozřejmostí je i ovládání animace jako je zastavení, posun zpět i dopředu, změna frekvence nebo pozastavení posledního snímku v animaci. Je to nepsaný standard, podle kterého se bude vytvářet i animace v této práci.

EUMETSAT

www.eumetsat.de

Oficiální stránky západoevropské mezivládní organizace EUMETSAT (European Organisation for the Exploitation of Meteorological Satellites). Na těchto stránkách lze najít snímky MSG v jednotlivých kanálech (interval 1 hodina) s možností volby oblasti, počtu snímků animace aj. a RGB kompozitní snímky MSG vzniklé kombinací několika kanálů (interval 1 hodina) s možností volby oblasti u některých produktů.

2.5 Flexové aplikace pro GIS

FlexGIS

FlexGIS je GIS Framework, který je zaměřen na jednoduchost a použitelnost, jak pro vývojáře, tak i pro uživatele. FlexGIS si klade za cíl přiblížit základní prostorové analýzy běžnému uživateli. Pilotní aplikace využívá Google Maps a prozatím toho moc nedokáže. Ovšem síla této aplikace spočívá ve snadnosti její rozšiřitelnosti. Tato aplikace, jak již název napovídá, je vytvořena v prostředí Adobe Flex, pomocí ActionScriptu a MXML.

Backend FlexGIS je navržen tak, aby byl co nejjednodušší, přijímá žádosti, vyřizuje je a vrací dat v použitelném formátu – typicky XML. Implicitně využívá PHP, MapServer a PostGIS

Více lze najít na www.flexgis.org, kde je i odkaz na samotnou aplikaci.

ArcGIS API pro Flex

Společnost ESRI zapracovala do svého ArcGIS serveru rozhraní API pro programování Flex aplikací. „Prostřednictvím API je možné například:

- Zobrazit interaktivní mapu, která bude obsahovat vaše data,
- spouštět GIS modely na serveru a zobrazovat výsledky,
- zobrazit svá data nad ArcGIS Online basemap,
- vyhledávat podle atributové složky vašich dat a zobrazovat výsledky,
- lokalizovat adresy a zobrazovat je,
- vytvářet mashup aplikace kombinací několika zdrojů dat.“ [Ožana: AIR, Flex a Flash: GIS do každého zařízení, Geobusiness 3/2009]

Poslední verzí ArcGIS API pro Flex je verze 1.2 z května 2009. Novinky této verze jsou v odkazu na <http://resources.esri.com/arcgisserver/apis/flex/>.

3 ÚVOD DO PROBLEMATIKY

3.1 Databáze

Pojem databáze dnes jistě není nikomu cizí. Celá dnešní moderní společnost je postavena na databázových systémech. Přímou či nepřímou s nimi pracujeme doslova denně. Jsou neodmyslitelnou součástí průmyslu informačních technologií.

3.1.1 Základní pojmy

Databáze – určitá uspořádaná množina informací (dat).

Data – libovolná posloupnost znaků, která může nést, ale nemusí, nějakou informaci.

Metadata – data o datech. Metadata v geoinformatice odpovídají na základní otázky „CO (název, popis a zařazení datové sady), KDO (původce, dodavatel dat), KDE (zeměpisné souřadnice, jména míst nebo oblastí), KDY (kdy byla datová sada vytvořena, jak často bude aktualizována, pokud vůbec), JAK (jak získat další informace, objednat datovou sadu, formáty, paměťová média, omezení)“ se stalo a děje ohledně nějakých dat [Ožana 2008].

Databázový systém – Database Management Systems (DBMS), v české literatuře označovaný jako systém řízení báze dat (SŘBD), slouží k uchování databází v paměti počítače. Tyto údaje umožňuje vkládat, opravovat, zobrazovat, třídit podle různých hledisek, tisknout v různých podobách, spojovat s jinými databázemi, kopírovat pro použití v jiných systémech.

Databázový server – vybavení (softwarové), na kterém je uložen databázový soubor a zároveň je zde spuštěn SQL server.

Databázový model – vyjadřuje způsob uspořádání dat v databázi. Je to v podstatě architektura, podle které databázový systém ukládá objekty do databáze a podle které je vzájemně provazuje. Umožňuje definovat schéma databáze, určující organizaci dat, způsoby jejich ochrany a zajištění správnosti (integritní omezení) a přípustné operace s daty. Podle typu modelovaných vztahů mezi záznamy v databázi se rozlišují:

- Otevřené soubory neboli souborové databáze.
- Hierarchický model.

- Síťový model.
- Relační model.
- Objektově orientovaný model.
- Objektově relační model.

Dotazovací jazyk – jazyk, který umožňuje ovládat databázi prostřednictvím příkazů – dotazů. Příkazy můžeme rozdělit na příkazy pro manipulaci s daty, příkazy pro definici dat a příkazy pro řízení dat. Příkladem dotazovacího jazyka je jazyk SQL, který patří v dnešní době k nejpoužívanějším. Dalšími příklady jsou: DMX, MDX, Datalog, MQL, OQL, SMARTS, QUEL.

3.1.2 Historie databází

Za předchůdce databází můžeme považovat papírové kartotéky, se kterými se na některých místech setkáváme dodnes. Umožňují uspořádat data podle různých kritérií a veškeré operace s nimi provádí přímo člověk. Prvním krokem k převedení zpracování dat na stroje se považuje sčítání lidu ve Spojených státech v roce 1890. Data byla zaznamenávána na dřevěný štítek a zpracování sebraných informací probíhalo na elektromechanických strojích, které se pro tyto účely využívaly dalších 50 let.

S vývojem počítačů v padesátých letech 20. století se objevil požadavek na vyšší jazyk pro zpracování dat. Výsledkem konference oborového sdružení Conference of Data Systems Languages (CODASYL) v roce 1960 byla publikovaná první verze jazyka COBOL, který se stal na další roky nejrozšířenějším jazykem pro hromadné zpracování dat. O pět let později na téže konferenci byla založena pracovní skupina Database Task Group (DBTG), která začala s návrhem množiny oborových standardů pro síťový model [Oppel, 2006]. Jedním z prvních průkopníků databází byl Charles Bachman.

Ve zprávě výboru z roku 1971 „The DBTG April 1971 Report“ se objevily pojmy jako schéma databáze, jazyk pro definici schématu, subschéma a podobně. Byla zde popsána celá architektura síťového databázového systému. Ve stejné době byly vyvíjeny i hierarchické databáze. Jedním z prvních SŘBD byl Information Management systém (IMS), který byl vyvinut firmou IBM pro program letu na Měsíc, Program Apollo. Systém IMS patří stále k nejrozšířenějším na sálkových počítačích.

„V červnu 1970 zveřejnil výzkumný pracovník společnosti IBM Dr. E. F. (Ted) Codd v odborném časopisu *Communications of the ACM, the Journal of the Association for Computing Machinery, Inc.*, článek s titulem „A Relational Model of Data for Large Shared Data Banks“ (Relační model dat pro velké sdílené data banky)“ [Oppel 2006: 32]. Tím začínají první relační databáze. Své standardy zveřejnila také skupina CODASYL DBTG. „Tím započalo pětileté období vášnivých sporů o to, který z modelů je nejlepší.

Zastánci síťového modelu CODASYL DBTG uváděli, že:

- Relační model je příliš matematický.
- Nelze vytvořit efektivní implementaci relačního modelu.
- Aplikační systémy potřebují zpracovávat data po jednotlivých záznamech.

Relační model obhajovali jeho tvůrci těmito argumenty:

- Něco tak komplikovaného, jako jsou návrhy DBTG nemůže být tou správnou cestou ke správě dat.
- Množinově orientované dotazy jsou v jazyku DBTG příliš obtížné.
- Síťový model nemá žádné formální základy v matematické teorii.

Tyto spory vyvrcholily roku 1975 na konferenci zvláštní zájmové skupiny ACM SIGMOD (Special Interest Group on Management of Data)“ [Oppel 2006: 32]. Ovšem argumenty obou stran byly naprosto správné, proto celé auditorium bylo po skončení konference ještě zmatenější než na jejím začátku. Koncem sedmdesátých let ale zájem o síťový model databází znatelně ochladl. Nakonec se, i díky dalšímu vývoji databází a počítačových technologií, ukázal relační model jako vhodnější [Oppel 2006].

Kolem roku 1974 se vyvíjí první verze dotazovacího jazyka SQL, který není tak přísně matematický. V 90. letech 20. století se objevovaly první objektově orientované databáze, které měly podle předpokladů vytlačit relační systémy. To se však nenaplnilo a vznikla kompromisní objektově-relační technologie, z níž jako první vstoupil na trh Informix a poměrně rychle za ním následovaly Oracle a IBM.

Ať už se jedná o snímky na ČHMÚ nebo na katedře aplikované geoinformatiky a kartografie při UK je jejich ukládání řešeno pomocí souborových databází. Pro účely této práce bude zvolena databáze relační.

3.1.3 Relační databáze

Výše zmíněná výzkumná práce Dr. E. F. Codd (viz kapitola 3.1.2), na jejímž základě se zrodil relační model databází, je považována za jeden z revolučních zlomů v historii vývoje počítačů.

Relace lze chápat jako tabulky dat uspořádaných do sloupců a řádků. Tyto tabulky jsou pak základem relačních databází. Jejich sloupce nazýváme atributy, řádky tabulky jsou záznamy. Atributy mají určen svůj konkrétní datový typ. Řádek je řezem přes sloupce tabulky a slouží k vlastnímu uložení dat. Konkrétní tabulka pak realizuje podmnožinu kartézského součinu možných dat všech sloupců – relaci.

V relačním modelu máme možnost svázat záznamy jen podle potřeby, nikoli podle vazeb definovaných předem při prvotním ukládání záznamů do databáze. V tomto modelu také pracují dotazy vždy s určitou množinou dat a ne s jednotlivými záznamy, jako tomu bylo u síťového a hierarchického modelu. [Oppel 2006].

Relační model je ze všech databázových modelů mezi moderními aplikačními systémy nejrozšířenější a to díky například těmto vlastnostem:

- „Definice, údržba i manipulace s datovými strukturami je snadná.
- Data je možné načítat prostřednictvím jednoduchých jednorázových dotazů.
- Data jsou v systému dobře chráněna.
- Existují dobře zavedené standardy ANSI (American National Standards Institute) a ISO (International Standardization Organization).
- Konkrétní produkty je možné vybírat od mnoha různých výrobců. Převody mezi různými implementacemi od různých výrobců jsou poměrně snadné.
- Produkty relačních databází jsou vyzrálé a stabilní“ [Oppel 2006: 34].

3.2 Rich Internet Applications - RIA

Samotný pojem RIA má několik významů. Původní zkratka znamená v překladu „bohatá internetová aplikace“. Význam RIA se ale posunul dále a dnes označuje směr, kterým se ubírají aplikace na internetu. Zkratka RIA dostala svůj význam Rich Internet Application poprvé v roce 2002 [Novák 2009]. „Technologie v této době postoupily a vyvinuly se tak daleko, že se přímo nabízel nový směr vývoje WWW aplikací“ [Novák 2009: 16]. RIA aplikace se dostali i na desktop a nesmí se zapomenout ani na masu mobilních zařízení, kde je zatím tato technologie především z výkonnostních důvodů v plenkách, ale nepotrvá dlouho a pojem RIA se bude skloňovat i zde. Oblasti, které RIA zhruba pokrývá, jsou zobrazeny na obr. 1.

Obr. 1: Hlavní oblasti obsahu RIA aplikací



Zdroj: [Novák 2009]

Rich Internet Applications jsou webové aplikace, které mají některé vlastnosti grafických desktopových aplikací. Díky mocným vývojovým nástrojům mohou být aplikace založené na RIA rychlejší a užitečnější. Uživatelé poskytnou lepší vizuální zážitek a vyšší interaktivitu oproti tradičním aplikacím, běžícím v prohlížeči, založeným na html a http. Příklady RIA jsou Adobe Flex/Flash, Ajax, Microsoft Silverlight a Java/JavaFX.

RIA se od tradičních webových aplikací liší v mnoha klíčových bodech. Jedním z nich je přímá interakce – RIA mohou využívat bohatší spektrum ovládacích prvků, než jen checkboxy, radio buttony a formulářová pole jako tradiční webové aplikace, a umožňuje tak vyšší efektivitu a lepší komfort pro uživatele (mohou například přímo editovat prvky stránky nebo je měnit pomocí drag-and-drop nástrojů). Dalším klíčovým bodem je obnovování části stránky. Tradiční aplikace nutí uživatele čekat, dokud se neaktualizuje celá stránka. Avšak RIA obsahují doplňkové technologie, jako real-time streamování, vysoce výkonné virtuální stroje běžící na straně klienta, a lokální cacheovací mechanismy, které dokáží snížit dobu čekání a zvýšit rychlost odpovědi. Díky RIA lze také mnohem efektivněji ovládat uživatelské rozhraní u různých prohlížečů a operačních systémů. Pokud jsou RIA navrženy tak, aby svůj stav ukládaly u klienta, je možné je využívat i pokud dojde k výpadku připojení k internetu [Novák 2009].

Z pohledu klient/server modelu řeší RIA čistě klientské záležitosti. Existence serveru se sice předpokládá, ale vůbec se neřeší, jak je serverová část naprogramována nebo na jaké platformě běží. Všechny RIA technologie jsou pouze o klientském programování.

Díky svým jedinečným vlastnostem – nezávislosti, možnosti práce offline, výkonu a dalším, bude tato technologie v geoinformatice hrát jistě důležitou roli.

„Výhodou je rovněž schopnost RIA technologie oprostit GIS od přílišné profesionality. Dokážou GIS zpřístupnit skutečně široké veřejnosti. Běžný uživatel, který chce využít GIS,

kvůli tomu nechce studovat pětiletý inženýrský obor. Chce například zjistit, zda v místě, kde plánuje koupit byt, není vysoká kriminalita. Nebo chce zjistit vzdálenost k nejbližší zastávce autobusu. Právě v těchto prostorových analýzách jim mohou pomoci chytré GIS aplikace a rostoucí popularita takových služeb je toho důkazem, viz například česká Lokola.“ [Ožana 2009: 37].

Využití popularity Flashe v geoinformatice je dáno možností spouštět Flash aplikace v mobilních zařízeních, dobrým zobrazením a zpracováním 3D dat a komunikací s periferiemi jako je kamera, fotoaparát a mikrofon, i s různými relačními databázemi. Na technologiích Flash je již dnes postaveno několik velmi úspěšných mapových služeb, např. Yahoo Maps nebo Google Maps for Flash. Adobe Flex své uplatnění již také pomalu nachází (viz kap. 2.5) [tamtéž].

3.3 Meteorologie

Vzhled oblohy a oblaků mohou vypovídat o právě probíhajících procesech v atmosféře. Často nám oblaka, jejich tvar, výška, množství a další ukazatele sdělují nejen diagnózu, ale i prognózu všeobecných meteorologických dějů. Samozřejmě na pozorování oblačnosti nemůžeme stavět seriózní předpověď počasí, avšak pro jistý úsudek o jeho vývoji na nejbližší hodiny nebo dokonce dny postačit může.

3.3.1 Základní pojmy

Oblak – „Neodborně mrak, mračno – viditelná soustava částic vody nebo ledu v atmosféře. Tato soustava může zároveň obsahovat i částice pocházející např. z průmyslových exhalátů, z kouře nebo prachu. Za oblak lze považovat i mlhu, která je v podstatě oblakem dotýkajícím se zemského povrchu. U oblaku se obvykle určuje: stupeň pokrytí oblohy oblaky čili oblačnost, optická hustota oblaků, druhy a tvar oblaků, výška základny oblaků, směr a rychlost tahu oblaků atd.“ [Bednář...(et al.) 1993: 199].

Oblačnost – „Stupeň pokrytí oblohy oblaky. Je důležitým meteorologickým prvkem, který nepřímo udává trvání slunečního svitu. Určuje se zpravidla odhadem. V synoptické meteorologii se vyjadřuje v osminách (viz kap. 4.3.2, tab. 2), v klimatologii v desetínách pokrytí oblohy oblaky. Nula znamená jasno, osm osmin, popř. deset desetin, zataženo“ [Bednář...(et al.) 1993: 199]. Termín polojasno se používá v případě, že převažuje řídká oblačnost, většinou oblaky středního a vysokého patra, jimiž prosvítá Slunce; pokud převládá hustší, zvláště kupovitá oblačnost, používá se označení polooblačno. Pojem oblačnost se také používá pro souhrnné, terminologicky ne zcela přesné označení pro skupinu určitých oblaků, např. oblačnost frontální, kupovitá, vrstevnatá, vysoká apod. [Bednář...(et al.) 1993].

Pozorování oblačnosti – „Vizuální sledování oblačnosti, popř. její sledování technickými prostředky (např. meteorologickou družicí, meteorologickým radiolokátorem), jehož výsledkem je stanovení druhů a dalších charakteristik oblaků podle mezinárodní klasifikace oblaků, celkového pokrytí oblohy, pokrytí jednotlivými význačnými druhy, výšky základny a tahu oblaků“ [Bednář...(et al.) 1993: 239].

Druhy oblaků – „Základní charakteristika oblaku podle mezinárodní klasifikace oblaků Světové meteorologické organizace. Vystihuje podstatné znaky vzhledu oblaku, které se jeví pozorovateli. Každý oblak, který se vyskytuje v troposféře, lze zařadit do jednoho z 10 druhů, a to cirrus, cirrocumulus, cirrostratus, altocumulus, altostratus, nimbostratus, stratocumulus, stratus, cumulus, cumulonimbus (více viz kapitola 4.3.2). Jeden a týž oblak nemůže současně náležet k dvěma druhům, tzn. Že se druhy u téhož oblaku vzájemně vylučují. Pro bližší popis oblaku se užívají další charakteristiky, tj. tvar oblaků, odrůda oblaků, zvláštnosti oblaků a mateřský oblak“ [Bednář...(et al.) 1993: 76].

3.3.2 Historie pozorování a klasifikace oblaků

Oblaky poutaly pozornost milovníků přírody i vědců odjakživa. Jako první poukázal na nutnost třídění oblaků známý francouzský přírodovědec Jean Baptiste Lamarck, který sám prováděl meteorologická pozorování a aktivně je propagoval. V ročence z roku 1802 publikoval svoji klasifikaci oblaků, ve které zavedl pět hlavních a několik doplňkových typů oblaků. Toto první třídění se však příliš neujalo [Skřehot 2008].

V roce 1803 se rozdělením oblaků a jejich systematickým názvoslovím proslavil amatérský meteorolog Luke Howard. Navrhl latinské názvosloví, a to pro tři hlavní druhy: cirrus, cumulus a stratus. Howardovo pojetí o oblacích našlo brzkou odezvu. V Praze se v polovině 19. století klasifikací oblačnosti zabýval rakouský fenolog a meteorolog Karl Fritsch, spolupracovník pražské Klementinské hvězdárny [Skřehot 2008].

V letech 1885 až 1887 H. M. Hildebrandsson a R. Abercromby navrhli skutečně moderní soustavu rozdělení oblaků. Jejich zásluhou došlo k mezinárodní dohodě a vydání prvního Atlasu věrných zobrazení tvarů oblaků, a tím i k jejich přesné definici. V letech 1896 a 1897 se pak uskutečnilo rozsáhlé mezinárodní pozorování oblaků prováděné podle jednotné metodiky, která byla přijata na mezinárodní meteorologické konferenci uspořádané v Paříži v roce 1896. Pozorování bylo doplněno i podrobným fotografováním, takže mohl být vydán první mezinárodní atlas typických oblakových tvarů [Skřehot 2008].

Za průkopníka fotografování oblaků v Československu a využití fotografií v meteorologii je považován dr. Antonín Bečvář, který se této činnosti systematicky věnoval ve čtyřicátých letech minulého století. Ve svém díle „Atlas horských mraků“, které vydal spolu s Bohuslavem Šimákem, popisoval vzhled oblaků na svých fotografiích i takovými latinskými přívlastky, které v současnosti již nejsou užívány. Dokonalost popisů ve své době neměla obdoby. Originální

jsou i filmy, v nichž Bečvář pomocí zrychleně promítaných snímků oblačnosti dokumentoval dynamiku oblačných dějů [Skřehot 2008].

Jednotný celosvětový systém klasifikace oblaků zavedla až Mezinárodní meteorologická organizace (WMO) se sídlem v Ženevě, která roku 1956 vydala „Mezinárodní atlas oblaků“. Ten je jako doporučení používán v globálním měřítku dodnes.

Moderní pojetí v přístupu k informacím dnes bezesporu představuje internet. V Evropě je nejznámějším a nejobsáhlejším elektronickým atlasem oblaků "Wolken Atlas" od Bernarda Mühra, který obsahuje stovky fotografií všech oblačných druhů, včetně jevů spojených s oblačností, a to ve velmi vysoké kvalitě, jež umožňuje právě digitální fotografie [Skřehot 2008].

K nejmodernějším tzv. distančním metodám pozorování stavu a vývoje oblačnosti náleží analýza družicových a radarových snímků.

3.3.3 Meteosat Second Generation – MSG

Jak již název napovídá, jde o druhou generaci družic Meteosat provozovaných evropskou organizací Eumetsat se sídlem v německém Darmstadtu. V praxi se družice také označují jako Meteosat 8. Vycházejí ze stejného konceptu jako Meteosaty první generace. Družice MSG jsou geostacionární, což umožňuje snímat stále stejné území v krátkých časových intervalech (15 minut) a díky následně vytvořeným animacím snímaných obrazových dat lze sledovat dynamiku vývoje oblačnosti nad daným územím.

Kromě snímání obrazových dat plní družice Meteosat několik dalších úkolů. Například sběr dat z automatických meteorologických stanic v nepřístupných oblastech, vyhledávání nouzových signálů a v neposlední řadě jsou také její pomocí dodávána základní meteorologická data méně vyspělým (zejména africkým) meteorologickým službám nebo letištím, která mají jinak omezený přístup k datům.

Charakteristikou je stabilizace na oběžné dráze pomocí rotace a současně využití rotace při pořizování snímků po řádcích. Doba rotace kolem své osy je 0,6 s, a při každé otočce se nasnímají tři řádky budoucího obrazu naráz. I s návratem zpět do výchozí polohy pak trvá cyklus snímání 15 minut.

Družice MSG nese několik různých přístrojů, pro meteorology je nejpodstatnějším přístroj SEVIRI, tedy dalekohled, v jehož ohniskové rovině jsou umístěna čidla citlivá na různé části spektra (jedná se vždy o trojici stejných čidel – tři řádky snímané naráz) [Charvát a Lacinová 2005].

Přístroj SEVIRI pořizuje snímky ve 12 spektrálních kanálech.

3.3.4 Distribuce meteorologických dat

Pojem data jsou v případě meteorologických družic myšlena jak data analogová, tak i digitální – družicové snímky. Ty jsou vytvářeny zaznamenáváním intenzity odraženého nebo emitovaného elektromagnetického záření objektů v určitých intervalech spektra. V České republice je největším odběratelem dat z meteorologických družic Český hydrometeorologický ústav. Sběrem, zpracováním a distribucí dat se zabývá družicové oddělení ČHMÚ sídlící na Libuši v Praze 4.

Data přijímá i katedra aplikované geoinformatiky a kartografie. Jejich zpracování pak probíhá v programu PCWINSAT-SG určeném pro data MSG.

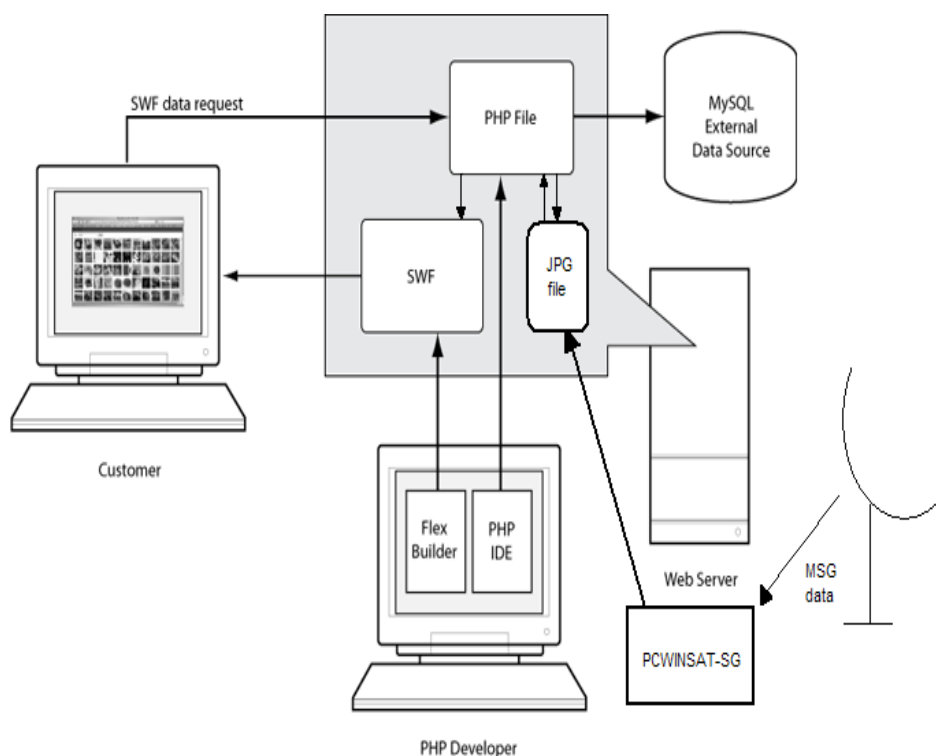
Data MSG jsou přijímána ve formátu označovaném HRIT – 12 spektrálních pásů každých 15 minut (METEOSAT) a ve formátu LRIT – 4 pásma každých 30 minut (GEOS, MTSAT). Data MSG jsou poskytována ve formátu označovaném LEVEL 1.5, což znamená, že data jsou korigována jak radiometricky, tak i geometricky pro polohu družice na 0°E/W a 0°N/S, doplněná o kalibrační a navigační informace.

Distribuce meteorologických dat je zajišťována pomocí systému EUMETCast, který je provozován organizací Eumetsat. Data z družice jsou nejprve předána do centrálního přijímacího střediska v Darmstadtu, kde jsou zpracována, doplněna o zeměpisnou síť, obrysy kontinentů, případně státní hranice a následně pomocí telekomunikační družice Eurobird 9 vysílána uživatelům.

4 PRINCIP TVORBY WEBOVÉHO PORTÁLU AKTUÁLNÍ METEOROLOGICKÉ SITUACE

Před samotnou tvorbou webového portálu je potřeba vytvořit návrh celé aplikace. Důležité je si uvědomit, co, jak, kdy a na co bude reagovat, jaké úkony se budou provádět, jaké možnosti má uživatel, jak a kde jsou uložena dat. Dílčím krokem při návrhu je vytvoření schématu celé aplikace. Schéma lze vidět na následujícím obrázku. Schématem bude postupně vysvětleno tak, jak se jím bude v průběhu práce postupovat.

Obr. 2: Schéma aplikace



Zdroj: vlastní

4.1 MySQL

MySQL je databázový server vytvořený švédskou komerční firmou MySQL AB, který se etabloval jako všudypřítomný server pro webové aplikace. Vychází, jak sám název napovídá, z programovacího jazyka SQL. Pro svou snadnou implementovatelnost, výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, trigger a uložené procedury [MySQL].

- V následující části je uveden výčet některých základních vlastností MySQL:
- Každá databáze v MySQL je tvořena z jedné nebo více tabulek, které mají řádky a sloupce.
- V řádcích rozeznáváme jednotlivé záznamy (řádek = záznam).
- Sloupce mají jméno a uvozují datový typ jednotlivých polí záznamu (sloupec = pole).
- Práce s databázemi, tabulkami a daty se provádí pomocí příkazů, resp. Dotazů.
- Systém MySQL je využitelný v C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic, .NET.

4.2 Návrh relační databáze

Návrh a implementace databází neprobíhá nikdy samostatně, spolu s databází se vyvíjí vždy další komponenty uceleného systému, jako je uživatelské rozhraní nebo aplikační programy. Veškeré události, které nastanou od okamžiku podnětu vzniku databáze, přes její vývoj a nasazení do provozu, lze shrnout do pojmu životní cyklus databáze. Zahrnuje etapy, kterými bychom při tvorbě databáze měli procházet – obr. 3.

Obr. 3: Životní cyklus databáze

Etapy	Aktivity v databázi
Plánování	Kontrola možností databázového systému
Sběr požadavků	Shromažďování a analýza názorů uživatelů. Identifikace předběžných entit.
Konceptuální návrh	Vývoj konceptuálního datového modelu.
Logický návrh	Vývoj logického datového modelu. Normalizace dat.
Fyzický návrh	Fyzický návrh databáze.
Konstrukční fáze	Vytvoření vývojových a testovacích databází. Testování potřebných konverzí dat.
Implementace a nasazení	Vytvoření provozních databází. Provedení potřebných konverzí dat.
Průběžná podpora	Vyladění výkonu databáze.

Zdroj: [Oppel 2006]

4.1.1 Plánování

První fáze procesu návrhu databáze je formulace úkolů a cílů. Je důležité si uvědomit, k čemu všemu bude databáze sloužit a jaká data bude shromažďovat. Potřebám pak návrh přizpůsobíme.

V přístupu řešení existují dvě metody – datový a funkční přístup. V datovém přístupu se klade důraz na analýzu dat a teprve podle výsledků se stanoví potřebné zpracování a transformace. Naopak při funkčním (procesním) přístupu se nejprve studují procesy a teprve poté stanoví, jaká data systém vyžaduje. Podle Oppela [Oppel 2006] praxe ukazuje, že nejlepší výsledky poskytuje kombinace obou přístupů. Jelikož ale na to není zpravidla dostatek času a lidí, přistupuje se k druhému nejlepšímu výsledku, a tím je datově orientovaný přístup.

Výsledkem této fáze by měl být dobře sestavený plán, do kterého bude možné nahlédnout v průběhu celého procesu návrhu databáze.

4.1.2 Sběr požadavků

Při sběru požadavků je důležité zaměřit se na to, **čeho** se má dosáhnout, nikoli **jak**. Do požadavků je třeba zahrnout co nejvíce informací o stávajících i očekávaných procesech, aplikačních pravidlech a entitách. Jako asi u většiny jiných procesů je i zde počáteční fáze velmi důležitá. Čím více práce se v této rané fázi projektu zvládne, tím hladší průběh budou mít následné etapy [Oppel 2006].

Mezi nejběžněji používané metody pro sběr požadavků patří rozhovory, dotazníky a pozorování.

Rozhovory jsou velmi rozšířenou metodou. Při diskuzi se často objeví postranní téma, které může být zdrojem dalších užitečných informací. „V přímém osobním rozhovoru je mnohem jednodušší odhalit nejistotu a pokusy o klamné odpovědi než v dotazníku s písemnými odpověďmi“ [Oppel 2006: 140]. Na druhou stranu tato metoda vyžaduje daleko více času i dostatečnou zkušenost tazatele než metody jiné.

Další metodou je **dotazník** s otázkami, který se rozešle všem rozhodovatelům a potenciálním uživatelům aplikace a databáze. Výhodou je malá časová náročnost, ovšem otázkou zůstává špatná návratnost dotazníku. Oppel [Oppel 2006] uvádí, že pokud uživatel není přinucen k odpovědi, ať už odměnou či hrozbou trestu, můžeme považovat za úspěch zhruba 10 % vrácených odpovědí.

Velmi rozšířenou technikou sběru požadavků je také **pozorování** činnosti osob, které budou s novou aplikací či databází pracovat. Při pozorování lze spatřit události či činnosti, které by lidé v odpovědi na dotazník nebo při rozhovoru nenapadlo nebo se neodvážili je uvést. Ale i tato metoda má své nevýhody, schované například v hawthornském efektu. Jde o to, že pokud lidé vědí, že jsou pozorováni, chovají se jinak a nelze tudíž získat přesný obraz o jejich pracovních procesech [Oppel 2006].

4.1.3 Konceptuální návrh

Úkolem fáze konceptuálního návrhu je návrh externího prostředí cílové aplikace a databáze. Konceptuální nebo také myšlenkový návrh znamená studium a modelování dat technologicky nezávislým způsobem. Výsledkem je ER diagram, neboli diagram entit a relací. Jeho položky (entita, atribut, relace) představují základní komponenty konceptuálního návrhu.

Entity jsou předměty z reálného světa, o kterých sledujeme údaje a zaznamenáváme je do databáze. Mohou jimi být osoba, místo, věc, událost, myšlenka apod., prakticky cokoli, co lze pojmenovat podstatným jménem. Entitu představuje v diagramu ER obdélník.

Atribut je vlastnost entity, kterou nějakým způsobem charakterizuje nebo popisuje. V ER diagramu jsou atributy znázorněny uvnitř obdélníku příslušné entity. Atribut, u kterého se vyskytuje symbol klíče, je jedinečný identifikátor entity, tzv. primární klíč. Každý atribut by měl být takový, aby jeho hodnotu nebylo možné dále rozdělit do několika menších jednotek, jinými slovy je atribut nejmenší pojmenovaná jednotka dat definovaná v databázovém systému. Někdy je ovšem potřeba řídit se i zdravým rozumem. Atribut adresa jde rozdělit do několika dalších atributů, otázkou je, do jaké míry je to pro dané účely smysluplné.

Protože v databázi jsou data, která spolu nějakým způsobem souvisí, je potřeba to v databázi zaznamenat. A k tomu slouží **relace**, které popisují vzájemné vztahy mezi entitami.

4.1.4 Logický návrh

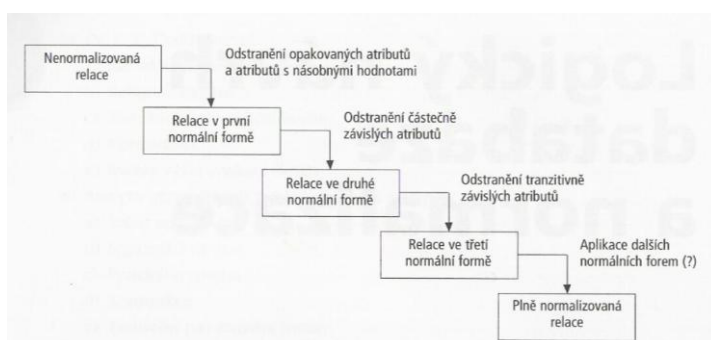
Během této fáze se provádí množství prací na technickém návrhu aplikace a databáze, navrhují vnitřní části projektu, se kterými uživatel nikdy nepřijde do styku.

„Z pohledu databáze je hlavním těžištěm prací této etapy takzvaná **normalizace**. To je technika návrhu tabulek pro relační databáze, kterou vyvinul Dr. E. F. Codd“ [Oppel 2006: 143]. Proces vyvinul v roce 1972 a navrhl v něm tři normální formy. Celý proces normalizace je znázorněn na obrázku 3. Na první pohled docela jednoduché, ale ve skutečnosti je potřeba velké množství praxe. Jde o odstranění opakujících se dat, omezení složitosti a zabránění tzv. aktualizacím anomáliím tak, aby se dospělo k přehlednější, rozšiřitelnější a výkonnější databázi.

Normalizace by měla vést ke vzniku tabulek, které lze snadno udržovat a efektivně se na ně dotazovat. Relace je v **první normální formě**, pokud každý její atribut obsahuje jen atomické hodnoty, tedy takové, které jsou z pohledu databáze dále nedělitelné. Relace se nachází ve **druhé normální formě**, jestliže je v první normální formě a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči. Druhá normální forma se tedy řeší jen v případě vícehodnotového primárního klíče. Ve **třetí normální formě** se nachází tabulka, která splňuje první dvě normy, a všechny její neklíčové atributy jsou navzájem nezávislé.

Od doby prvotního zavedení normalizace se řada autorů pokusila o zavedení dalších verzí. Příkladem je **Boyce-Coddova normální forma**, kdy v relaci nesmí existovat žádný určující atribut, který by byl primárním klíčem nebo kandidátním klíčem. To znamená, že žádný neklíčový atribut nesmí jedinečně identifikovat hodnotu žádného jiného atributu, a to ani atributů, které se účastní definovaného primárního klíče. Pokud je tabulka v Boyce-Coddově formě a popisuje pouze příčinnou souvislost, nachází se ve **čtvrté normální formě**. Ta se zabývá vztahy uvnitř složeného primárního klíče. Vyžaduje, aby klíč tvořily jen ty hodnoty, které mají skutečnou vzájemnou souvislost. A nakonec **pátá normální forma**, ve které je tabulka, která je ve čtvrté normální formě a není možné do ní přidat další atribut tak, aby se vlivem skrytých závislostí rozpadla na několik dílčích relací [Oppel 2006].

Obr. 3: Proces normalizace dat



Zdroj: [Oppel 2006]

4.1.5 Fyzický návrh

Fyzický návrh znamená implementaci normalizovaných relací do prostředí konkrétního relačního databázového systému. Celý proces Oppel [Oppel 2006] popisuje takto:

1. Z každé normalizované relace vytvoříme tabulku.
2. Z každého atributu v normalizované relaci vytvoříme sloupec odpovídající tabulky. Pro každý sloupec musíme určit:
 - Jedinečný název sloupce v rámci tabulky. Většinou se převezmou názvy atributu z logického návrhu, někdy je však nutné název změnit.
 - Datový typ a pro některé datové typy i jeho délku.
 - Zda jsou hodnoty sloupců povinné nebo ne. Tato podmínka se v daném sloupci promítá do klauzule NULL nebo NOT NULL.
3. Z jedinečného identifikátoru relace vytvoříme primární klíč tabulky (sloupec musí mít NOT NULL).
4. Pro každou další skupinu sloupců, které mají být v tabulce jedinečné, můžeme definovat jedinečné omezení.
5. Ze vztahů mezi normalizovanými logickými relacemi se ve fyzickém návrhu stávají referenční omezení.
6. Rozsáhlé tabulky je vhodné rozdělit, pokud tedy použitý relační databázový systém dělení podporuje. Výrazem dělení se označuje funkce databázového systému, která umožňuje rozdělení jedné tabulky do několika fyzických komponent a uložení těchto komponent do samostatných datových souborů, přičemž celé dělení je pro databázového uživatele transparentní (tabulka se mu jeví jako jediná, nerozdělená).

Provádí se zde také předběžná analýza potřebných databázových dotazů.

4.2 Adobe Flex

Jak již bylo zmíněno v kapitole 3.2, existuje pro tvorbu RIA aplikací několik technik. Zatím u nás nejméně prozkoumaná a využívaná, i ve spojení využití v geoinformatice, je Adobe Flex. A právě proto byla tato technika zvolena pro potřeby této práce.

Flex, pocházející z dílny Adobe, je vyspělá technologie mající za sebou mnoho let vývoje. Pomocí této technologie můžeme vytvářet jak RIA, tak i klasické desktopové aplikace. Velikou výhodou je, že aplikace vytvořené pomocí Flexu běží ve Flash Playeru, který je již dnes

nainstalován na drtivě většině počítačů připojených k internetu, a to díky možnosti vytvořit soubor typu *.swf* přímo ze zdrojového kódu. Adobe Flex tedy „umožňuje takové RIA aplikace, které zkompilují do stejného formátu jako aplikace Adobe Flash a běží ve stejném běhovém prostředí (navíc umožňuje spouštění i v běhovém prostředí Adobe AIR)“ [Novák 2009]. Dá se říci, že Adobe Flex si bere z Adobe Flash to nejlepší a snaží se přidávat nové vlastnosti tam, kde u Adobe Flash chyběly.

Framework Adobe Flex zahrnuje podporu efektů a animací, které lze přiřadit jednotlivým prvkům aplikace – např. filtry rozmazávání, lupa, různé přechody při zobrazování prvků, sekvence pohybů, změny velikosti.

Pro vývoj ve Flexu se používají dva jazyky. Prvním z nich je mxml, který se podobá html. Mxml primárně slouží pro tvorbu grafického rozhraní aplikace. Je to jazyk postavený na xml, který je intuitivní a snadno čitelný.

Dalším, již plnohodnotným, programovacím jazykem je actionscript, v aktuální verzi 3. Tento jazyk, jako většina, vychází syntaxí z jazyka C a dobře kombinuje rysy javascriptu a javy. Actionscript můžeme použít i pro programování vzhledu, ovšem v praxi se používá spíše pro programování samotného fungování aplikace, tedy pro aplikační logiku.

Flex obsahuje předem sestavenou knihovnu tříd a aplikačních služeb, jako je „navázání dat, řízení přetažení, zobrazovací systém, který řídí rozvržení uživatelského rozhraní, systém stylů, které řídí vzhled komponent rozhraní, a efekty a systém animací, který řídí pohyb a přechody“ [Armsoft].

Díky novému běhovému prostředí Adobe AIR lze aplikace vytvořené ve Flexu spustit nejen v prohlížeči, ale také instalovat na desktop. Flex je stále aktivně vyvíjen, má kolem sebe poměrně širokou komunitu lidí, existuje pro něj řada komponent a open source projektů. Vedle AJAXu je dnes nejpoužívanější RIA technologií ve světě. Tam, kde už nestačí klasické metody pro vývoj webových aplikací pomocí html, css nebo javascriptu, je vhodné použít právě Flex.

4.2.1 Adobe Flex Builder 3

K vývoji Flex aplikací vydalo Adobe balík Flex Builder, aktuálně verzi 3. Flex Builder obsahuje kvalitní IDE postavené na Eclipse, kompilátor, debugger a samotný framework. Studenti si tento balík mohou stáhnout zdarma, ostatní mohou využít šedesátidenní zkušební verzi. Shrnutí informací o Adobe Flex Builderu zobrazuju tab. 1.

Tab. 1: Adobe Flex Bulder – základní informace

Adobe Flex Builder	
Vývojář	Adobe Systems (dříve Macromedia)
Aktuální verze	3.3.0 (3. dubna 2009)
OS	Windows, Mac OS X, Linux
Typ softwaru	Interaktivní tvorba webu, vektorová grafika, programování v ActionScript, MXML
Licence	open source
Web	http://adobe.com/flex

Zdroj: [Adobe]

Na rozdíl od programu Adobe Flash, ve Flex Builderu máme spoustu již hotových komponent (objektů) s vlastní grafikou, vlastnostmi a metodami, které nám ušetří spoustu práce. Práce s Flex Builderem je rychlá.

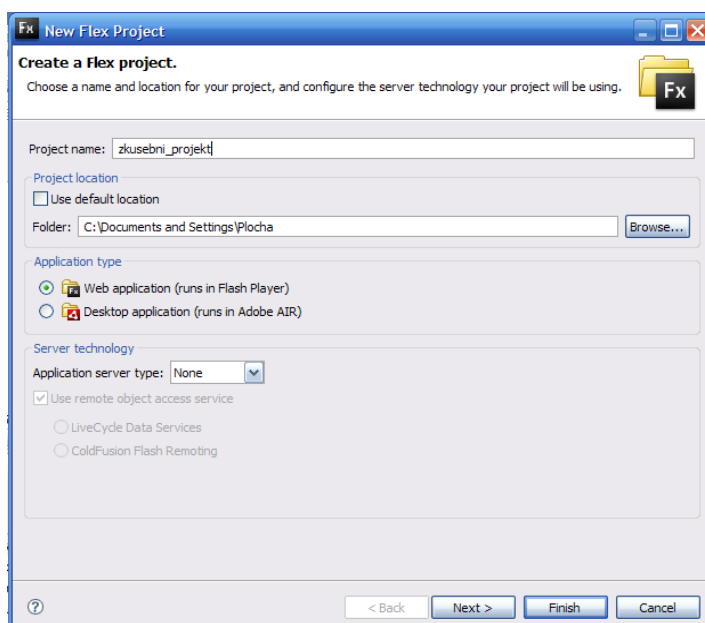
Produkt je dostupný v angličtině a japonštině. Požadavky na systém Windows uváděné na oficiálním webu [Adobe] jsou následující:

- Procesor Intel Pentium 4.
- Systém Microsoft Windows XP s aktualizací Service Pack 2 nebo Windows Vista Home (Premium nebo Basic), Business nebo Ultimate.
- 1 GB paměti RAM (doporučeny 2 GB).
- 500 MB volného místa na pevném disku (pro konfiguraci modulů plug-in je vyžadováno dalších 500 MB).
- Prostředí Java Virtual Machine: Sun JRE 1.4.2, Sun JRE 1.5 (obsaženo), IBM JRE 1.5 nebo Sun JRE 1.6.
- Software Eclipse 3.2.2, 3.3 a 3.4 pro konfiguraci modulů plug-in (pro systém Windows Vista je doporučována verze Eclipse 3.3).
- Přehrávač Adobe Flash Player 9.
- Aplikace BEA Workshop 10.1.
- Nástroj IBM Rational Software Architect 7.0.0.3 (pouze konfigurace modulu plug-in Eclipse 3.3).

4.2.2 Vytvoření projektu v Adobe Flex

Proces vytváření aplikace v Adobe Flex začíná založením projektu a pokračuje vývojem až ke spuštění a otestování. Nový projekt se vytváří prostřednictvím File > New > Flex projekt (obr. 4). Při zakládání je samozřejmostí pojmenování a umístění projektu. Dále je možno si zvolit, zda se má vytvořit webová či desktopová aplikace.

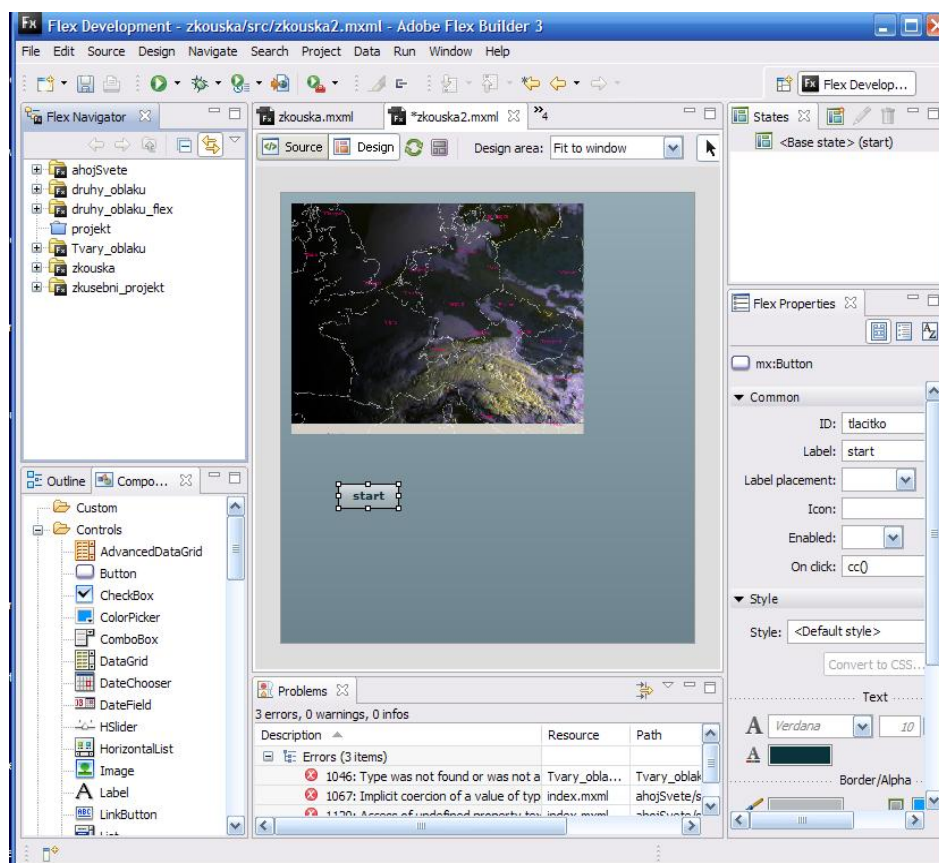
Obr. 4: Screenshot vytvoření projektu



Zdroj: vlastní

Spolu s projektem se otevře i nový MXML soubor pojmenovaný stejně jako projekt, čehož si lze všimnout v levém okně File navigátor (obr. 5). Do souboru se postupně umísťují grafické komponenty, jako jsou tlačítka, obrázky apod., přetažením z levého dolního okna nebo zapsáním přímo v kódu (obr. 5), editace totiž probíhá ve dvou režimech – designerském a režimu psaní samotného kódu. Vpravo se nachází okno Flex properties, kde jsou vypsány nejdůležitější vlastnosti vybrané komponenty (v případě obr. 5 to je id tlačítka, nápis na tlačítku, možnost úpravy stylu písma, akce reagující na kliknutí tlačítka atd.)

Obr. 5: Screenshot vzhledu aplikace



Zdroj: vlastní

S uložením aplikace dojde i k jejímu přeložení do SWF. Ve File navigátoru, ve složce bin-debug se vytvoří soubory .swf a .html, tudíž není potřeba vytvářet HTML soubor, ani provádět kompilace aplikace. Soubor .swf je klasický soubor spustitelný ve Flash Playeru a .html pak tento SWF zobrazí.

4.2.3 MXML a actionscript

Soubor typu .mxml je XML soubor, jenž umožňuje deklarativní tvorbu aplikací. Ta je podobná tvorbě WWW aplikací pomocí HTML dokumentů. „Ve skutečnosti je formát MXML pouze prvek, který usnadňuje programování Adobe Flex aplikací. Struktura a logika aplikace zapsané pomocí MXML souborů je posléze převáděna do formátu ActionScript. Vše, co lze vytvořit pomocí MXML souborů, lze vytvořit také pomocí ActionScript kódu“ [Novák 2009]. I když je zápis uživatelského rozhraní pomocí MXML jednodušší a přehlednější, je asi nejjednodušší forma tvorby aplikací kombinovat oba přístupy.

Základní deklarace MXML je:


```
<?xml version="1.0" encoding="utf-8"?>

<mx:Application                                xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute">

</mx:Application>
```

Jak lze z kódu vidět, celou aplikaci uvozuje komponenta **application**. Podobně tomu je i při zápisu **ActionScriptu**, kdy je kód uvozen komponentou **Script**:

```
<mx:Script>

<![CDATA[

] >

</mx:Script>
```

Soubory **ActionScript** se také dají psát i samostatně do souborů formátu **AS** a následně do aplikace importovat.

4.2.5 Flex a php

PHP se obvykle používá ve spojení s **HTML** a **javascriptem**. Lze jej ale snadno integrovat i s **Flexem** a vytvořit tak dynamičtější rozhraní. **SWF** vložený do webové stránky provádí úkony u klienta. **PHP** komunikuje přímo s databázovým serverem. **Flex** si ale data bere z webové služby, protože s databází přímo nekomunikuje, a odtud je zobrazí.

Při práci se skripty **PHP**, které pracují s **MySQL**, se používá koncept klient/server. Jde o to, že na serveru běží naprosto samostatně databázový stroj **MySQL**, který čeká na příkazy ze strany skriptu **PHP**. Pokud chceme pracovat s daty, musíme komunikovat s databázovým strojem **MySQL**. Vlastní komunikace z hlediska skriptu **PHP** vypadá takto:

1. Připojení se k databázovému serveru **MySQL**,
2. Zaslání příkazů **SQL** databázovému serveru **MySQL**,
3. Převzetí výsledných dat z databázového serveru **MySQL**,
4. Odpojení se od databázového serveru **MySQL**.

4.3 Práce se snímky MSG

4.3.1 Zpracování dat

Zpracování dat MSG použitých v této práci má na starosti program PCWINSAT-SG. Jde o komerční produkt pro zpracování surových dat EUMETCast. Je dílem společnosti UKWtechnik sídlící v německém Baiersdorfu, která se specializuje na satelitní příjem meteorologických dat [Eumetsat].

PCWINSAT-SG běží na operačním systému MS Windows XP (případně 2000) a skládá se z pěti modulů. **DVB Reception** kontroluje příjem dat, **TelliCast Decoder** vypisuje data obdržená v závislosti na licenci, která pak průběžně skenuje modul **MSG-Splitter**. Přichází data jsou řazena podle uživatelem definovaných kritérií a jsou uložena v příslušných adresářích na disku. **MSG-Preprocessing** poté podle definice uživatele generuje požadované produkty. Zvolit si lze spektrální kanály a zeměpisnou oblast zájmu. Poslední modul **PCWINSAT-SG Image Processing** zobrazí hotové produkty (obrázky), umožní jejich další úpravu, jako změnu barev, projekce, histogramu a další, a samozřejmě export ve standardních grafických formátech.

4.3.2 Klasifikace oblačnosti

Díky družicovým snímkům mohli meteorologové celého světa spatřit oblačnost i z vnější strany troposféry. Ze snímků si mohli ověřit své představy o vývoji tlakových níží a front. Dnes kromě obrázků oblačnosti poskytují meteorologické družice také informace o vertikálním rozložení teploty vzduchu, o dlouhovlnném záření zemského povrchu a spodních vrstev atmosféry, o teplotě vody na hladině oceánů, o rozložení sněhové pokrývky, o rozsahu záplav na řekách, o sopečných výbuších apod.

Vybraný produkt VIS-IR kombinuje spektrální kanály HRV a IR 10.8. Barevná kombinace je blízká vnímání lidským okem. Nízká až střední oblačnost je zobrazena nažloutle, vysoká oblačnost bíle až modře. Terén, pokrytý vegetací má zelenou barvu a voda tmavě modrou. Pro toto barevné zobrazení bude klasifikace snímku v této práci zaměřena na rozlišení oblačnosti.

Záběry z družic prozradí, jak je oblačnost v jednotlivých oblastech mohutná a jak rychle se až dosud přemísťovala. O jejím dalším vývoji by už měli vypovídat matematické modely.

Mezinárodní klasifikace oblaků rozlišuje deset základních druhů, které se orientačně rozdělují podle výšek, v nichž se v našich zeměpisných šířkách nejčastěji vyskytují:

1. **Výšky 6-9 km, popř. i výše, v zimě naopak někdy níže:** cirrus, cirrocumulus, cirrostratus.
2. **Výšky nejčastěji 2-5 km nad terénem:** altostratus, altokumulus.
3. **Oblaky se spodní základnou obvykle do výšky 2 km nad terénem:** stratus, stratokumulus, nimbostratus.
4. **Oblaka s vertikálním vývojem:** cumulus, cumulonimbus.

Kromě deseti základních druhů rozlišuje Mezinárodní klasifikace oblaků ještě tvary, odrůdy, zvláštnosti a tzv. mateřské oblaky [Bednář 2003].

Podrobnosti lze najít v Mezinárodním atlase oblaků a také na webových stránkách k této práci: <http://geo.natur.cuni.cz/hrstkova>.

Děje formulující počasí i sám vývoj oblaků jsou velice složité, proto na otázku, zda nám pohled na oblaky může něco říci o počasí v nejbližších hodinách, neexistuje odpověď ve formě jednoduchého pravidla. Ovšem vzhled oblohy, respektive oblačnost, kterou na ní lze pozorovat, může o dějích v atmosféře mnohé napovědět. „O ustáleném, pěkném počasí, zpravidla v létě v oblastech vysokého tlaku vzduchu, svědčí typický denní chod kupovité oblačnosti, kdy k ránu a k večeru máme jasno, zatímco od pozdních odpoledních hodin se objevují cumuly s nepříliš velkým vertikálním vývojem (tzv. oblaky pěkného počasí)“ [Bednář 2003: 104].

„Typickou předzvěstí blížící se fronty s jejími počasovými projevy jsou cirrovité oblaky, a to především tehdy, jestliže se jejich množství na obloze v čase rychle zvětšuje, houstnou-li a obsahují-li vlákna s háčky zahnutými nahoru (tzv. cirrus uncinus – obr. 6)“ [Bednář 2003: 104]. Háčky či fajfkovitá zahnutí u těchto oblaků znamenají, že vítr ve výškách nad zhruba 7 až 8 km téměř jistě přesahuje rychlost 40 m/s a že ovzduší tam je poměrně turbulentní. Při zemi přitom ještě vůbec nemusí foukat silný vítr. Přejedem fronty se také všeobecně zhorší počasí. Cirry jsou na frontu vázány hlavně v zimě, v létě mohou naopak indikovat stálé a postupně čím dál teplejší počasí a to vyskytují-li se na čisté obloze, nepřibývají a nehoustnou [Dvořák 2001].

Obr. 6: Cirrus uncinus



Zdroj: [Dvořák 2001]

Cirrocumulus, jehož vzezření připomíná velké množství malých cumulů, indikuje labilitu ve vysokých vrstvách troposféry, společně s dostatečnou zásobou vlhkosti. Pokud se v letním období přidá ještě mohutná tepelná energie, lze očekávat v dohledné době vývoj tzv. bouřek z tepla, zvláště v horách [Dvořák 2001, Roth 2000].

Přítomnost cumulu, oblaka spodního patra troposféry, značí existenci vertikálních pohybů. Cumuly lze rozdělit do tří skupin: ploché, středně vyvinuté a vysoko vyvinuté (věžovité). Ploché cumuly (cumulus humilis – obr. 7) se také lidově nazývají „oblaka pěkného počasí“ a opravdu jsou průvodním jevem krásného, letního a suchého počasí. Středně vysoké cumuly (cumulus mediocris) bývají na letní obloze často po přechodu studené fronty ze západního sektoru. Věžovitý cumulus (cumulus congestus – obr. 7) je už znamením velké lability v troposféře a velmi často předzvěští bouřku, zvláště vyskytuje-li se už časně dopoledne [Dvořák 2001, Roth 2000].

Obr. 7: Tvary oblaku cumulus

Cumulus humilis



Cumulus congestus



Zdroj: [Dvořák 2001]

Z družicového snímku lze určit i pokrytí oblohy oblačností, i když smysl to má pouze nad menším územím. Zjištění pokrytí je i jedním z cílů této práce. Pokrytí oblohy oblačností se udává v synoptice v osminách a v klimatologii v desetínách. Následující tabulka vyjadřuje pokrytí oblohy v osminách a tato klasifikace bude použita i u snímků pro webový portál.

Tab. 2: Pokrytí oblohy oblačností

Pokrytí	
0/8	jasno
1/8	skoro jasno
2/8	skoro jasno
3/8	polojasno
4/8	polojasno
5/8	oblačno
6/8	oblačno
7/8	skoro zataženo
8/8	zataženo

Zdroj: [Dvořák 2001]

5 POSTUP TVORBY WEBOVÉHO PORTÁLU AKTUÁLNÍ METEOROLOGICKÉ SITUACE

5.1 Vytvoření databáze

Samotné snímky budou uloženy v adresáři. V databázi se uloží pouze metadata a cesta ke snímkům. Podle požadavků uživatelů byla navržena databáze, která následně prošla etapami vývoje, popsány v kapitole 4.1.1 – 4.1.5.

Databáze bude shromažďovat metada o družicových snímcích pro potřeby webového portálu. Těmito dvěma základními požadavky se řídí již samotný návrh databáze. Projevila se zde kombinace obou přístupů řešení, a to datového i funkčního. Z předpokládaných procesů se stanovilo, jaká data systém vyžaduje, a zároveň se analyzovala data, která lze o snímku získat. Již od začátku se tedy uvažovalo o procesech i samotných datech.

Pro získání požadavků byla zvolena metoda rozhovorů, které byly prováděny mezi potencionálními uživateli. Zde je třeba připomenout, že je portál určen pro „běžného“ uživatele. Data by měla poskytovat jasnou a srozumitelnou informaci i pro neodborné publikum. Při rozhovorech byly zjišťovány potřeby uživatelů, ale i jejich znalost s interpretací družicových snímků, nebo spokojenost se stávajícími systémy.

Ukázalo se, že nejvyhledávanější a také nejjednodušeji čitelný je produkt snímků MSG **VIS-IR**. K popularizaci a užitečnosti tohoto produktu přispěla hlavně média, která jej používají a předkládají tak uživatelům. Dále vynikla potřeba možnosti vidět čas i datum pořízení snímku přímo při animaci. Častým přáním pak byla jistá předpověď meteorologické situace ze snímků, což je ovšem nad rámec této diplomové práce.

Po analýze všech těchto získaných informací se mohlo přistoupit ke konceptuálnímu návrhu.

V této fázi probíhá studium a modelování dat technologicky nezávislým způsobem. Výsledkem je diagram entit a relací. Jednotlivé položky diagramu představují základní komponenty konceptuálního návrhu, jsou jimi entita, a tribut a relace. Pro potřeby této práce byly navrženy entita **snímky**, která uchovává data o družicových snímcích a entita **oblačnost** s typy pokrytí oblohy oblačností. Zvolena byla pak pouze entita snímky, která informaci o pokrytí oblohy oblačností nese v sobě. Použít prvotní návrh by chybou nebylo, ovšem po zvážení náročnosti

výkonu a potřeb změn popisu pokrytí oblohy oblačností bylo rozhodnuto popisovat pokrytí oblohy přímo v entitě snímků.

Nyní nastává čas pro práci na technickém návrhu aplikace a databáze. Podle kapitoly 4.1.4 je hlavním těžištěm této fáze takzvaná normalizace. Lze ji popsat jako soubor pravidel, podle kterých by se mělo postupovat při transformaci struktury entit a relací ER modelu na strukturu fyzického uspořádání tabulek a relací v databázi. Protože se v tomto případě relace nevyskytují, nezabere proces normalizace příliš práce. Uvádí se, že třetí normální forma pokryje více než 90 % všech případů a proto se považuje za jakýsi „zlatý standard“. Ani v tomto případě tomu nebude jinak.

Po dokončení fáze logického návrhu přechází projekt do fáze fyzického návrhu. Prvním krokem je promítnutí normalizovaných relací do skutečných, „fyzických“ tabulek. Tento proces je popsán v kapitole 4.1.5. Tabulka nesou stejný název jako entita. Z atributů byly vytvořeny sloupce tabulek, opět stejně pojmenované. Každému sloupci je dále potřeba přiřadit datový typ. Jedná se o jakési kategorie formátu konkrétního sloupce. Datové typy omezují množinu povolených dat ve sloupci jen na takové znaky, které mají pro daný datový typ smysl, a pomáhají relačnímu databázovému systému efektivně ukládat data sloupce. Obrázek 9 ukazuje definici tabulky **Snímky**. Datový typ každého ze sloupců je uveden ve druhém sloupci zleva.

Obr. 8: Definice tabulky Snímky

	Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra	Akce							
<input type="checkbox"/>	<u>cesta</u>	varchar(20)	utf8_czech_ci		Ne	Žádná									
<input type="checkbox"/>	datum	datetime			Ne	Žádná									
<input type="checkbox"/>	hodnota_nizka	int(3)			Ano	NULL									
<input type="checkbox"/>	hodnota_vysoka	int(3)			Ano	NULL									
<input type="checkbox"/>	nizka	varchar(20)	utf8_czech_ci		Ano	NULL									
<input type="checkbox"/>	vysoka	varchar(20)	utf8_czech_ci		Ano	NULL									

Zdroj: vlastní

Z jedinečného identifikátoru relace byl vytvořen primární klíč tabulky. V tabulce Snímky jím je cesta. Pro primární klíč, stejně jako pro sloupce, které mají být v tabulce jedinečné, se definují jedinečná omezení.

Každá relace mezi entitami z konceptuálního návrhu se v logickém návrhu promítá do referenčního omezení, což je takové omezení, které zajišťuje či vynucuje relaci mezi tabulkami. Databázový systém pak může automaticky kontrolovat, zda ke každé hodnotě cizího klíče v závislé tabulce existuje odpovídající hodnota v řídící tabulce. V tomto případě se cizí klíče nevyskytují, neboť se jedná pouze o jednu tabulku.

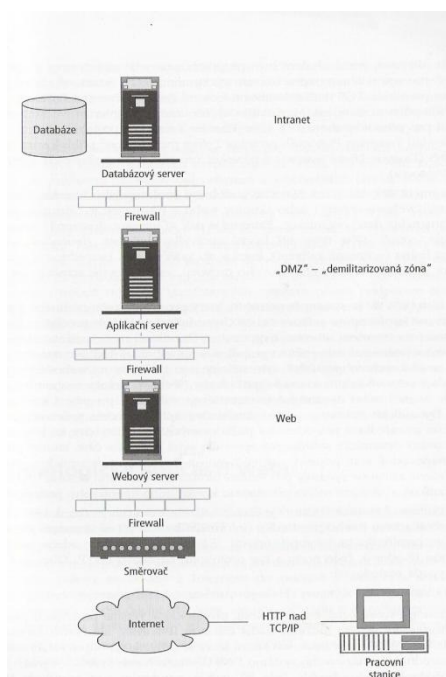
Součástí fyzického návrhu je také předběžná analýza potřebných databázových dotazů. Je třeba otestovat, zda pomocí příkazů lze získat data, potřebná pro webový portál. Jedná se zejména

o datum, čas a typ pokrytí oblačností, ale také o informaci o cestě ke snímku důležitou pro skript php.

5.2 Připojení databáze k webu

Pro provoz aplikačního systému a odpovídající databáze na Internetu je potřeba poměrně dost různých technologických vrstev. Základní komponenty celého uspořádání jsou zobrazeny na obr. 10. Na pracovní stanici klienta běží webový prohlížeč komunikující po Internetu v protokolu HTTP nad TCP/IP. Webový server je umístěn za tzv. směrovačem, který předává datové pakety mezi jednotlivými sítěmi, a za firewallem a je zodpovědný za umístění webových stránek a jejich předkládání po přijetí požadavku. Zpracování adres URL na webovém serveru si někdy vyžádá spuštění příslušné transakce na aplikačním serveru. Jestliže daný požadavek vede k načtení dat z databáze, odesílá aplikační server do databázového serveru příslušný dotaz [Oppel 2006].

Obr. 10: Databáze připojené k webovému prostředí



Zdroj: [Oppel 2006]

Pro správu databázového serveru MySQL a pro práci s daty se už tradičně využívá nadstavba zvaná **phpMyAdmin**. Je to už dlouhá léta vyvíjený nástroj, který umožňuje vykonávat všechny

základní akce nad databázovým serverem MySQL, jako jsou vytváření nebo rušení databáze, vytváření, úprava nebo rušení tabulek a provádění SQL příkazů.

5.2.1 PHP

O ukládání dat do databáze i jejich příjem z ní se postará PHP skript. Pomocí něj lze zasílat SQL příkazy do databáze, ukládat nová a mazat stará data. O způsobu komunikace PHP s MySQL bylo již zmíněno v kapitole 4.2.5.

Jelikož jsou snímky umístěny v adresáři, je třeba uložit jejich metadata do databáze. Metodou **opendir** () lze celý adresář otevřít a metodou **readdir** () přečíst názvy všech souborů v něm. Jelikož se do databáze ukládá cesta a datum (ve formátu *YYYYMMDDhhmmss*), je potřeba ještě před uložením tyto proměnné z názvu vytvořit. Následující kód se postará o otevření adresáře a přečtení souborů:

```
$adresar = dir("msg");  
  
$soubor = $adresar->read();
```

V adresáři se bude uchovávat stále stejný počet souborů – snímků (292). Jakmile bude do adresáře vložen nový snímek, nejstarší se z adresáře vymaže. Tento proces je postaven na porovnání datumu nejmladšího snímku v databázi s datem snímku v adresáři. Pokud se v adresáři vyskytne nějaký mladší, uloží se metadata do databáze, ze které se zároveň metadata nejstaršího snímku odstraní.

Ještě než se metadata uloží, proběhne snímek funkcí na zjištění pokrytí oblačností. O tomto bude zmíněno až v kapitole 5.4.

Aby se údaje o snímcích dostali do databáze v co nejkratší době od jejich pořízení, bude skript PHP listovat v adresáři každých 15 minut počínaje celou hodinou. V tomto intervalu jsou snímky pořizovány družicí.

Úplný zdrojový kód skriptu si lze prohlédnout v příloze 1 na konci této práce.

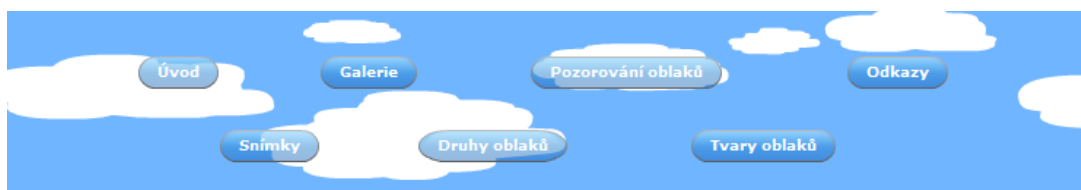
5.3 Webový portál

5.3.1 Vzhled portálu

Pro vytvoření layoutu stránek byla zvolena metoda kaskádových stylů – CSS. Oproti jiným metodám umožňuje oddělení obsahu od vzhledu, kód je přehlednější a stručnější, díky němuž se načítá rychleji. Stránka vytvořená tímto způsobem se zobrazuje postupně s tím, jak se načítá. Díky tomu může uživatel vidět na svém monitoru reakci prohlížeče téměř okamžitě po kliknutí. Nevýhodou ovšem je neúplná podpora ze strany prohlížečů. Různé brokery podporují různé množiny vlastností CSS různým způsobem, a neexistuje žádný prohlížeč, který by je podporoval přesně podle norem W3C.

Portál byl navržen tak, aby se v něm uživatel snadno orientoval, nemusel procházet složitými cestami a aby jej na první pohled dokázal zaujmout. Jak již bylo několikrát zmíněno, pomáhají tomu i jazyky mxml a actionscript. Vzhled menu portálu je zobrazen na následujícím obrázku, lze jej také vidět přímo na <http://geo.natur.cuni.cz/hrstkova/>.

Obr. 10: Výběrové menu portálu



Zdroj: Vlastní

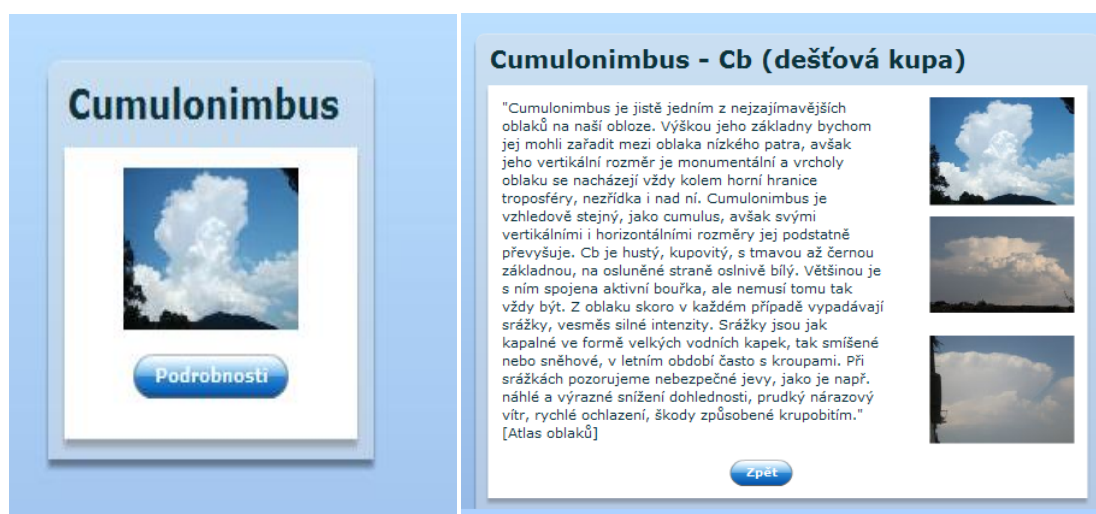
Pozadí menu tvoří animace, které byl vzorem příklad na serveru Flash.cz – serveru pro kreativní lidi. Animaci netvoří jeden dlouhý objekt, jak by se mohlo zdát, ale pouze jeho malá část, která se podle potřeby opakuje. Procesor tak není tolik zatěžován. Objektem je možné pohybovat v obou směrech a to pomocí kurzoru myši. Zdrojový kód je k nahlédnutí opět v příloze – „Animace pozadí“.

Menu rozdělí portál na několik částí: úvod, animaci snímků, malou encyklopedii oblaků a odkazy na webové stránky a literaturu s podobnou tematikou. Technické řešení jednotlivých částí bude vysvětleno v následujícím textu.

Úvodní stránka přivítá uživatele snímkem střední Evropy s pásmem oblačnosti táhnoucí se od západu k východu a textem informujícím o tom, co vše díky meteorologickým družicím pozorovat a o produktu MSG vybraném pro animaci. To vše je zasazeno do krátké animace, která by měla uživatele upoutat a pobídnout k delšímu setrvání na těchto stránkách.

Malá encyklopedie oblaků má za úkol poskytnout uživateli základní informace o druzích, tvarech a pozorování oblaků. Aby se encyklopedie nestala pouhým výčtem informací, byly použity efekty vytvořené ve Flex Builderu. Druhy oblaků jsou rozděleny na čtyři sekce dle dělení podle výšky základny na oblaky vysoké, středního a nízkého patra a oblaky přesahují hranice pater. V každé sekci jsou pak krátký text o jednotlivých zástupcích spolu s fotografiemi. Zde bylo použito efektu změny stavů. Nadefinován byl jeden základní a několik doprovodných, které byli vyvolány stiskem tlačítka. Vzhled základního i doprovodného stavu si lze prohlédnout na obr. 11 nebo na adrese http://geo.natur.cuni.cz/hrstkova/druhy_oblaku.html.

Obr. 11: Změna stavu



Zdroj: vlastní

Část věnovaná tvarům oblaků je vytvořena v objektu Accordion. Jedná se o navigátor „kontejnerů“, které zobrazuje jako řadu panelů pod sebou. Výhodou je, že lze všechny panely zobrazeny najednou a uživatel jimi prochází podle jejich hlavních složek. Na dalších řádcích je zobrazen ukázkový kód z knihy *Getting Started with Flex 3* [Herrington, Kim 2008]:

```
<mx:Accordion id="ac"
    width="335"
    selectedIndex="0"
    historyManagementEnabled="false">
    <mx:VBox label="Yummy!"
```

```

        horizontalAlign="center"

        verticalAlign="middle"

        width="300" height="300">

        <mx:Image source="assets/dessert_decadent_cake.jpg" />

        <mx:Button label="Order - $5.00"

            click="ac.selectedIndex=5" />

    </mx:VBox>

</mx:Accordion>

```

Zdrojový kód použití objektu Accordion k prezentaci tvarů oblaků je v příloze této práce.

Část věnovaná odkazům by se zdála být stránkou s pouhým výčtem odkazů a literatury. Ale i pro tuto část byl ve Flex Builderu vytvořen efekt, a to efekt „rostoucího“ obrázku. V odkazech na literaturu s podobnou tematikou jsou zobrazeny údaje o knize spolu s obrázkem přední části obalu. Právě tento obrázek při najetí kurzoru myši jakoby vyroste. Inspirací byl opět příklad Herringtona a Kima [Herrington, Kim 2008]. Jedná se o jednoduchý efekt, který přiblíží vybraný objekt.

Nejdůležitější části stránek, animaci snímků, je věnována následující kapitola.

5.3.2 Animace snímků

Animace snímků byla provedena pomocí jazyků actionscript a mxml. Jedná se o funkci, která prochází polem vybraných snímků a po jednom je zobrazuje v animaci. Po několika testech bylo zjištěno, že nejlepší metodou přenosu proměnných je zde přenos přes URL – viz následující kód:

```

imgLoader.contentLoaderInfo.addEventListener(Event.COMPLETE,
loadComplete);

imgLoader.load(new
URLRequest("http://geo.natur.cuni.cz/hrstkova/"+imagesURL[0]));

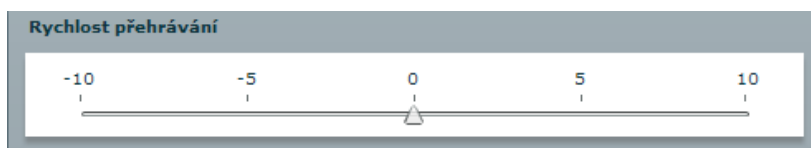
```

PHP skript, který se stará o načítání údajů o snímcích z databáze, zobrazí díky metodě GET hodnoty do URL adresy. Protože těmi hodnotami jsou cesty ke snímkům, zobrazuje actionscript snímky přímo z adresy URL. Přenos je poměrně rychlý, i když vše závisí na počtu

vybraných snímků. S vyšším počtem se rychlost načítání samozřejmě snižuje. Při porovnání s jazykem javascript, je rychlost načítání přijatelná i při výběru větším než 20 snímků. Pokud jsou snímky načteny, může se spustit samotná animace. Nejprve ale něco k jejímu ovládání.

Do selectového pole se zobrazuje datum všech snímků v databázi, vždy od nejaktuálnějšího k nejstaršímu. Počet položek je dán výběrem snímků při ukládání do databáze. Poté, co uživatel označí zvolené snímky a nahraje pomocí buttonu „nahraj“ do animace, může ji plně ovládat k tomu určeným prvkem a tím je hslider (obr. 12). Pro ovládání byl zvolen pouze tento jeden prvek. Lze jím animaci spustit, zastavit, zrychlit, zpomalit a to ve směru tam i zpět. Při výběru prvků ovládání mezi hsliderem nebo několika klasickými buttony byl zvolen pro větší jednoduchost a pohodlí uživatele právě hslider. Jedním ovládacím prvkem tak lze ovládat několik funkcí.

Obr. 12: Ovládací prvek animace - hslider



Zdroj: vlastní

Z délky kódu samotné animace a jejího ovládání (kódy viz níže) lze usoudit rozdíl množství práce při použití actionscriptu a javascriptu, kde by byl kód mnohem delší.

```
public function on_timer(e:Event) {  
    if(vpravo) {  
        _img.getChildAt(cisloObr).visible=false;  
        cisloObr+=1;  
        if(cisloObr==_img.numChildren) cisloObr=0;  
        _img.getChildAt(cisloObr).visible=true;  
    }else{  
        _img.getChildAt(cisloObr).visible=false;  
        cisloObr-=1;  
        if(cisloObr==-1) cisloObr=_img.numChildren-1;  
    }  
}
```

```

        _img.getChildAt(cisloObr).visible=true;
    }
}

public function changeSlider():void{

    vpravo = slider.value>0 ? true : false;

    if(myTimer!=null){

        myTimer.stop();

        if(slider.value != 0){

            myTimer.delay = 500 / Math.abs(slider.value);

            myTimer.start();

        }

    }

}

```

Animace je doplněna komentářem stavu pokrytí snímku oblačností. Uživatel si může nechat zobrazit u vybraného snímku procentuální vyjádření pokrytí nízkou i vysokou oblačností i slovní komentář. Zjištěním takového pokrytí se zabývá následující kapitola.

Na výběr je i možnost zobrazit si přes snímky základní GIS vrstvy – hranice, vodstvo, sídla a komunikace. GIS vrstvy byly zkonvertovány do průhledného formátu png a tento je zobrazován nad snímky v animaci podle zvoleného výběru.

5.4 Klasifikace oblačností

Jedním z cílů této práce je klasifikovat oblačnost ze snímku. Vzhledem k vybranému produktu se jedná o určení rozsahu pokrytí oblohy nízkou a vysokou oblačností. Nízká a vysoká oblačnost je rozlišena vizuálně podle barvy na snímku. Oblaka nízkého až středního patra jsou zobrazena nažloutle a vysokého patra bíle až modře.

Rozsah pokrytí oblačností se určuje podle rozsahu výskytu pixelů s příslušnou barvou. K tomu byla použita funkce PHP **imagecreatefromjpeg**. Původně byl řešen návrh, že by tato analýza byla prováděna actionscriptem. Protože jsou ale snímky ukládány na serveru, a výstup z analýzy v databázi, nemá analýza na straně klienta, kterou by actionscript prováděl, smysl. Výsledek by

se tak musel posílat zpět na server. Proto bylo přistoupeno k PHP, které analyzuje snímek přímo na serveru a to vždy s příchodem nového snímku (viz kap. 5.2.1).

Analýzou několika zkušebních snímků byly zjištěny hodnoty RGB výskytu nízké a vysoké oblačnosti. Tyto hodnoty pak byly převedeny na barevný model HSV (hue, saturation, brightness). Tento barevný model nejvíce odpovídá lidskému vnímání barev. Sestává se ze tří složek:

- **Hue** – barevný tón neboli odstín (0° až 360°).
- **Saturation** – sytost barvy. Představuje množství šedi v poměru k odstínu. Měří se v procentech od 0 % (šedá) do 100 % (plně sytá barva).
- **Value** – hodnota jasu. Relativní světlost nebo tmavost barvy.

HSV se pokouší popisovat vnímání barevných vztahů přesněji než RGB, přesto zůstává výpočtově jednoduchý.

Skript převádí hodnoty RGB na snímku na HSV:

```
function hsv($red, $green, $blue){  
  
    $r = $red / 255.0;  
  
    $g = $green / 255.0;  
  
    $b = $blue / 255.0;  
  
    $H = 0;  
  
    $S = 0;  
  
    $V = 0;  
  
    $min = min(min($r, $g), $b);  
  
    $max = max(max($r, $g), $b);  
  
    $delta = $max - $min;  
  
  
    $V = $max;  
  
    if($delta == 0){  
  
        $H = 0;  
  
        $S = 0;}  
  
    else{
```

```

$S = $delta / $max;

$dR = ((($max - $r) / 6) + ($delta / 2)) / $delta;

$dG = ((($max - $g) / 6) + ($delta / 2)) / $delta;

$dB = ((($max - $b) / 6) + ($delta / 2)) / $delta;

if ($r == $max)

    $H = $dB - $dG;

else if($g == $max)

    $H = (1/3) + $dR - $dB;

else

    $H = (2/3) + $dG - $dR;

if ($H < 0)

    $H += 1;

if ($H > 1)

    $H -= 1;}

```

Skript prochází pixel po pixelu a kontroluje, zda hodnota spadá do předem stanovených intervalů. Pokud ano, přičte pixel do součtu a nakonec vyrobí procentuální poměr. Tyto hodnoty jsou uloženy do databáze spolu se slovním hodnocením podle synoptické meteorologie. Uživatel si pak tyto hodnoty a komentář může nechat kdykoli zobrazit.

6 DISKUZE

Publikování dat v prostředí internetu je dnes považováno za samozřejmost. Ani geodata nesmějí zůstat pozadu. Ale k tomu, aby poskytla uživatelům nějaké informace je třeba je vhodně prezentovat. Pojem RIA – Rich Internet Application je dnes stále populárnější. Tato technologie poskytuje možnosti, které jsou daleko za pouhé čtení. Objevují se i první náznaky využití této technologie v geoinformatické. Tato diplomová práce je dalším z nich. Snaží se navrhnout prezentaci dat DPZ uživatelům pomocí technologie, která umožňuje efektivněji ovládat uživatelské rozhraní a zlepšit komfort pro uživatele.

Cílem této práce je tedy navrhnout webový portál aktuální meteorologické situace s využitím družicových dat. V těchto zeměpisných šířkách se podobných portálů moc nevyskytuje. Jediným hlavním konkurentem a zároveň inspirací je animace snímků na portále českého hydrometeorologického ústavu. Z důvodu širších možností se s ním nelze srovnávat v obsahové stránce informací. Úkolem bylo, vytvořit příjemné uživatelské prostředí a to nejen k animaci, ale i k ostatním částem portálu.

V této práci byl použit pouze jediný produkt MSG snímků, který kombinuje spektrální kanály HRV a IR 10.8. Díky této kombinaci ze snímku „vystupuje“ nízká a vysoká oblačnost, podobně jako při pouhém pozorování oblačnosti ze země. A protože oblaka upoutávají pozornost většiny uživatelů snad denně, byl vybrán právě produkt, který je tomuto pozorování nejbližší. Tato vizuální klasifikace snímku je doplněna o vyjádření rozsahu pokrytí oblačností a to jak číselně, tak i slovně. I když k přesnému rozboru stavu počasí a jeho dalšího vývoje je potřeba i dalších metod, může oblačnost o dějích v atmosféře uživateli mnohé napovědět. Pokud se k tomu přidá i určitá znalost atmosférických jevů, lze současný stav ovzduší a jeho vývoj odhadovat i z množství oblaků a jejich rozmanitosti. Tím se naplňuje hypotéza o vhodnosti vybraného produktu a jeho automatické klasifikace pro přehlednou výpověď o stavu oblačnosti.

Portál byl vytvořen v poměrně novém prostředí Adobe Flex. V diplomové práci Nováka [Novák 2009], ve které prováděl analýzu RIA metod a technik, se Adobe Flex umístil na prvním místě. To vypovídá o tom, že se jedná o opravdu silnou technologii. Společnost Adobe je také jednou z nejdůležitějších společností v oblasti RIA technologií. To zřejmě plyne z produktu Adobe Flash, který stál u vzniku celé oblasti RIA aplikací a je velmi rozšířená a známá. Na jedné stránce nabízí Adobe Flash dobré vlastnosti pro RIA aplikace, ovšem vývoj pomocí ní nenabízí

dostatečné možnosti. Proto je výhodnější pro vývoj použít framework Adobe Flex. Takovýto postup byl zvolen i v této práci.

Hlavní částí v této práci vytvořeného portálu je animace snímků MSG. Ta byla vytvořena jako flash animace a dále upravena v prostředí Adobe Flex, kde byli také vytvořeny i ostatní části portálu. Tato animace a její ovládání může plně konkurovat animaci na portálu čhmú, vytvořené v javascriptu. Na většině webových prohlížečích je javascript pomalejší a ani nenabízí příliš vstřícné vytvoření tak jednoduchého a příjemného ovládání. Animace i ostatní informativní části portálu splňují svůj účel vhodné prezentace snímků MSG. Proto lze konstatovat, že se potvrdila i druhá hypotéza. Adobe Flex je vhodný pro tvorbu informačního portálu s využitím družicových dat.

Práce by měla být přínosem těm, kdož chtějí efektivně prezentovat svá data na internetu a nespokojí se s dosud používanými metodami.

7 ZÁVĚR

Úvod této práce se zaměřuje na teoretický pohled na danou problematiku. Pozornost je věnována všem vědním disciplínám, do kterých práce zasahuje, tedy databázím, RIA technologiím a meteorologii.

V další části práce byl proveden návrh ukládání a manipulace se snímky. Vytvořená databáze snímků je specifická pro potřeby informačního portálu. Snímky jsou proto ukládány na serveru a v databázi jsou pouze jejich metadata. Načítání je tak mnohem rychlejší. Zvoleno bylo vytvoření pouze jedné tabulky, ve které jsou shromažďovány o snímcích veškeré potřebné informace pro použití na webovém portálu. Ukládání nových dat do databáze, stejně jako odstranění starých, je samozřejmě řešeno automaticky a to s periodou příchodu nových snímků z družice – 15ti minut.

Posléze šlo o návrh automatické tvorby animace ze snímků, která je základem webového portálu navrženého v prostředí Adobe Flex. Samotná animace byla vytvořena v prostředí Adobe Flash a upravena v Adobe Flex. Uživatel si může zvolit libovolný počet snímků uložených v adresáři, které se pak nahrají a zobrazují v animaci.

Do portálu pak byly začleněny i doprovodné funkce. Jednalo se o malou encyklopedii oblaků vytvořenou pomocí několika efektů, tak aby podávala uživateli informace tou nejpříjemnější cestou. Řešena byla i integrace GIS vrstev v rámci této aplikace.

V neposlední řadě byl navržen způsob automatické klasifikace oblačnosti na snímku. Jedná se o určení rozsahu pokrytí oblačností. Právě v tomto kroku je spatřen možný další vývoj, který je nad rámec této práce. Automatická klasifikace by zřejmě šla zpřesnit a rozšířit na možnost klasifikovat pouze vybranou část snímku nebo získat jiné informace, než jen rozsah pokrytí oblačností. S tím souvisí i možné zpřesnění krátké předpovědi vývoje počasí. Jak ale již bylo zmíněno, na tuto oblast v práci již nezbyl dostatek prostoru.

Aplikace je určena pro „běžného“ uživatele internetu, kterému by měla ve vstřícné formě zprostředkovat data DPZ, jejich přínos a být i impulsem možného hlubšího zájmu.

SEZNAM POUŽITÝCH ZKRATEK

apod.	a podobně
atd.	a tak dále
cit.	citováno
ČHMÚ	český hydrometeorologický ústav
MSG	Meteosat Second Generation
např.	například
obr.	obrázek
SQL	Structured Query Language
str.	strana
tab.	tabulka
tzn.	to znamená

SEZNAM OBRÁZKŮ, TABULEK A PŘÍLOH

Seznam obrázků

1	Hlavní oblasti obsahu RIA aplikací	str. 18
2	Schéma aplikace	str. 23
3	Životní cyklus databáze	str. 25
4	Screenshot vytvoření projektu	str. 31
5	Screenshot vzhledu aplikace	str. 32
6	Cirrus uncinus	str. 36
7	Tvary oblaku cumulus	str. 36
8	Definice tabulky Snímky	str. 39
9	Databáze připojené k webovému prostředí	str. 40
10	Výběrové menu portálu	str. 42
11	Změna stavu	str. 43
12	Ovládací prvek animace – hslider	str. 45

Seznam tabulek

1	Adobe Flex Bulder – základní informace	str. 30
2	Pokrytí oblohy oblačností	str. 37

Seznam příloh

- 1 Zdrojový kód vybraných částí aplikace
- 2 CD s elektronickou verzí práce. Html, php a swf soubory

SEZNAM ZDROJŮ INFORMACÍ

ARMSOFT: [online]. [cit. 3.8.2009]. Dostupné z URL <<http://www.amsoft.cz/produkty/adobe/flex>>.

ADOBE: [online]. [cit. 20.6.2009]. Dostupné z URL <<http://www.adobe.com>>.

BEDNÁŘ, J. (2003): *Meteorologie*. 1. vydání, Portál, Praha, 224 str. ISBN: 80-7178-653-5.

BEDNÁŘ, J....[et al.] (1993): *Meteorologický slovník výkladový a terminologický*. 1. vydání, Ministerstvo životního prostředí ČR, Praha, 594 str. ISBN: 80-85368-45-5.

ČHMÚ: [online]. [cit. 13.7.2009]. Dostupné z URL <<http://www.chmi.cz>>.

DVOŘÁK, P. (2001): *Atlas oblaků*. 1. vydání, Svět křídel, Cheb, 128 str. ISBN:80-86808-42-0.

EUMETSAT: [online]. [cit. 13.7.2009]. Dostupné z URL <<http://www.eumetsat.de>>.

FLEXGIS: [online]. [cit. 24.7.2009]. Dostupné z URL <<http://www.flexgis.org>>.

HERRINGTON, J.; KIM, E. (2008): *Getting Started with Flex 3*. 1. vydání, Adobe Developer Library, 143 str. ISBN: 0-596-52064-6.

CHARVÁT, Z., LACINOVÁ, M. (2005): *Země z vesmíru*. [elektronický zdroj]. Praha: MFF UK. Dostupné jako soubor PDF z URL <<http://www.mff.cuni.cz/verejnost/mgalerie/ARCHIV/meteosat2005/meteosat2005.pdf>>.

KERK, M. (2008): *Design and integration of a EUMETCast receiving installation*. [elektronický zdroj]. Stuttgart: Berufsakademie, Applied Computer Science. Dostupné jako soubor PDF z URL <http://www.lehre.dhbw-stuttgart.de/~sto/public/stud_arb/2007/Eumetcast.pdf>.

MYSQL: [online]. [cit.]. Dostupné z URL <<http://www.mysql.com>>.

NOBLE, J.; ANDERSON, T. (2008): *Flex 3 Cookbook: Code Recipes, Tips, and Tricks for RIA Developers*. 1. Vydání, Adobe Developer Library, 704 str. ISBN: 0596529856.

NOVÁK, J. (2009): *Analýza RIA metod a technik* [rukopis]. Brno, 121 str. Diplomová práce na fakultě informatiky Masarykovy univerzity. Vedoucí diplomové práce RNDr. Jan Pavlovič.

OPPEL, A. (2006): *Databáze bez předchozích znalostí*. 1. Vydání, Computer Press, Brno, 320 str. ISBN: 80-251-1199-7.

OŽANA, R. (2009): *AIR, Flex a Flash: GIS do každého zařízení*. *Geobusiness*. 2009, roč. 8, č. 3, str. 36-37. ISSN: 1802-4521.

OŽANA, R. (2008): *Velký přehled databází*. *Geobusiness*. 2008, roč. 7, č. 11, str. 45-46. ISSN 1802-4521.

SKŘEHOT, P. (2008): *Velký atlas oblaků*. 1. vydání, Computer Press, Brno, 368 str. ISBN: 978-80-251-2015-6.

ROTH, G., D. (2000): *Encyklopedie počasí*. 1. vydání, Euromedia Group, Praha, 296 str. ISBN: 80-242-0228-X.

WARD, J.; COENRAETS, CH.; WILSON, G. (2009): *Tour de Flex*. [online]. Dostupné z URL < <http://www.adobe.com/devnet/flex/tourdeflex>>.

PŘÍLOHA 1 – ZDROJOVÝ KÓD APLIKACE

Styl CSS

```
/* CSS Document */

body { margin:0; padding:0; background:#FFEEED; text-align:center; }

#obsah {width: 755px; margin:1em auto; text-align:left; border: 1px
solid #228; }

#levy {width: 755px; height: 440px; float: left; background-color:
#ADD1FF ; border-left: 1px solid #228; border-right: 1px solid #228;}

#levy_text {text-align: left; padding: 1px 2em; scrollbar-face-color:
green; width: 690px; height: 420px; overflow:auto;text-indent: 40px;
}

#levy_text_uvod {text-align: center; padding: 1px 2em;}

#levy_text h2{text-align: center; padding: 1px 2em;}

#levy_text h3{text-align: left; font family: 'Times New Roman', Times,
serif; font weight: bold; padding: 1px 2em;}

#levy_text h4{text-align: left; font family: 'Times New Roman', Times,
serif; font weight: bold; padding: 1px 2em;}

#levy_text_mapa{text-align: center; padding: 1px 2em;}

#paticka { padding: 6px 4px 2px 4px; border: 1px solid #228;
background: #8F66FF; }
```

Úvod

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

  <head>

    <link rel="shortcut icon" href="favicon.ico">

    <meta http-equiv="content-type" content="text/html; charset=windows-
1250">

    <meta name="generator" content="PSPad editor, www.pspad.com">
```

```

<title>Úvod</title>

<link href='styl.css' rel='stylesheet' type='text/css' />

</head>

<body>

<div id='obsah'>

    <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        id="index" width="755" height="132"

        codebase="http://fpdownload.macromedia.com/get/flashplayer/current/swflash.cab">

            <param name="movie" value="index.swf" />

            <param name="quality" value="high" />

            <param name="bgcolor" value="#869ca7" />

            <param name="allowScriptAccess" value="sameDomain" />

            <embed src="index.swf" quality="high"
bgcolor="#869ca7"

                width="755" height="132" name="index"
align="middle"

                play="true"

                loop="false"

                quality="high"

                allowScriptAccess="sameDomain"

                type="application/x-shockwave-flash"

                pluginspage="http://www.adobe.com/go/getflashplayer">

            </embed>

        </object>

<div id='levy'>

<div id='levy_text'>

<br>

```

```

<br>

<div id='levy_text_mapa'>



</div>

<br><br>

<p>text<p>

<p>text</p>

<p></p>

<p>text</p>

</div>

</div>

<div id="paticka">

    <table>

    <tr>

    <td class='ikona'><a href="http://www.chmi.cz/">

        </a></td>

        <td class='copy'>&copy; 2009, Lucie Hrstková</td>

    </tr>

    </table>

    </div>

</div>

</body>

</html>

```

Animace pozadí

```

posuv = 5;

pressed = false;

startX = obloha._x;

this.onEnterFrame = function():Void {

    if (pressed) {

        posuv = (_root._xmouse-mouseX)/2;}

        obloha._x -= posuv;

        if (obloha._x<=(startX-obloha._width/2)) {

            obloha._x = startX;

        } else if (obloha._x>=startX) {

            obloha._x = startX-obloha._width/2; }

        if (Math.abs(posuv/1.1)>5) {

            posuv = posuv/1.1; }

    };

this.onMouseDown = function():Void {

    posuv = 0;

    pressed = true;

    mouseX = _root._xmouse;

};

this.onMouseUp = function():Void {

    pressed = false;

};

```

Animace snímků

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute"
        applicationComplete="InitApp(event)" height="500" width="755">

<mx:Script>

```

```

<![CDATA[
    import mx.events.DragEvent;
    import flash.external.*;
import flash.utils.Timer;
import flash.events.TimerEvent;
import flash.display.Bitmap;
import flash.events.MouseEvent;

    public var imagesURL:Array;
    public var images:Array = new Array();
    public var myTimer:Timer;
    public var imgLoader:Loader = new Loader();

    public var i:int;
    public var cisloObr:int = 0;
    public var vpravo:Boolean = true;
    var r1:RegExp = /seznam./g;
    var r2:RegExp = /seznam%5B%5D=/g;
    var r3:RegExp = /%2F/g;

    public function InitApp(e:Event):void{
        //slider.addEventListener(MouseEvent.MOUSE_MOVE,
sliderChange);
        mujtext.text += "čekejte, prosím, ";
        var _queryString =
ExternalInterface.call("window.location.search.substring", 1);
        mujtext.text += "snímky se načítají, "+_queryString +
"\n";
        if(_queryString==null) mujtext.text= "nelze načíst
PHP proměnné (IE)";

        else {
            _queryString = _queryString.replace(r1, "");
            _queryString = _queryString.replace(r2, "");
            _queryString = _queryString.replace(r3, "/");
            imagesURL = _queryString.split('&');

            if(imagesURL[0].length==0) {imagesURL = [];
mujtext.text="Vyberte snímky"}
            else{

                imgLoader.contentLoaderInfo.addEventListener(Event.COMPLETE,
loadComplete);

                imgLoader.load(new
URLRequest("http://geo.natur.cuni.cz/hrstkova/"+imagesURL[0]));
                mujtext.text+="Načítám:";

            }

        }

    }

    public function loadComplete(e:Event):void {
        _img.addChild(e.currentTarget.content);
        mujtext.text+="\nNačten "+int(cisloObr+1)+" . obrázek.
"

        cisloObr+=1;
        if (_img.numChildren != imagesURL.length)
imgLoader.load(new
URLRequest("http://geo.natur.cuni.cz/hrstkova/"+imagesURL[cisloObr]));

```

```

        else {
            mujtext.text+="Obrázky načteny!";
            mujtext.visible=false;
            myTimer = new Timer(200);
            myTimer.addListener(TimerEvent.TIMER,
on_timer);

            cisloObr=0;
        }
    }
    public function on_timer(e:Event){
        if(vpravo){
            _img.getChildAt(cisloObr).visible=false;
            cisloObr+=1;
            if(cisloObr==_img.numChildren) cisloObr=0;
            _img.getChildAt(cisloObr).visible=true;
        }else{
            _img.getChildAt(cisloObr).visible=false;
            cisloObr-=1;
            if(cisloObr== -1) cisloObr=_img.numChildren-1;
            _img.getChildAt(cisloObr).visible=true;
        }
    }
    /*public function playStop(e:MouseEvent):void{
        trace(ps.label)
        if(ps.label=="Stop"){
            if(myTimer!=null) myTimer.stop();
            ps.label="Play";
        }else{
            if(myTimer!=null) myTimer.start();
            ps.label="Stop";
        }
    }
    */
    public function changeSlider():void{
        vpravo = slider.value>0 ? true : false;
        if(myTimer!=null){
            myTimer.stop();
            if(slider.value != 0){
                myTimer.delay = 500 /
Math.abs(slider.value);
                myTimer.start();
            }
        }
    }
}
]]>
</mx:Script>

<mx:UIComponent x="0" y="0" width="37" height="33" id="_img"/>
<mx:TextArea x="208" y="317" width="339" height="75"
id="mujtext"/>
<!--
<mx:Button x="587" y="702" label="Přehrát vpřed &gtt;"
click="{vpravo=true; myTimer.start();}"/>
<mx:Button x="408" y="702" label="&lt;t; Přehrát zpět"
click="{vpravo=false; myTimer.start();}"/>
<mx:Button x="527" y="702" label="Stop" id="ps"
click="playStop(event)"/>
-->

```

```

        <mx:Panel x="132" y="400" title="Rychlost přehrávání"
verticalAlign="middle" horizontalAlign="center" width="491"
height="90">
            <mx:HSlider x="408" y="623" id="slider" minimum="-10"
change="changeSlider()"
                maximum="10"
                tickInterval="5"
                liveDragging="true"
                dataTipPrecision="0"
                snapInterval="1"
                labels = "{[-10, -5, 0, 5, 10]}"
                allowTrackClick="false"
                showDataTip="true"
                value="0"
                width="428" height="44"/>
        </mx:Panel>

</mx:Application>

```

Změna stavu – druhy oblaků, část „oblaka přesahující hranice pater“

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" height="436"
width="650" cornerRadius="13" horizontalAlign="center"
verticalAlign="middle" backgroundColor="#ADD1FF" layout="absolute">
    <mx:states>
        <mx:State name="podrobnosti">
            <mx:SetProperty target="{okno}" name="width"
value="518"/>
            <mx:SetProperty target="{okno}" name="height"
value="396"/>
            <mx:SetProperty target="{okno}" name="x"
value="66.5"/>
            <mx:SetProperty target="{okno}" name="y" value="20"/>
            <mx:SetProperty target="{tlacitko}" name="x"
value="201"/>
            <mx:SetProperty target="{tlacitko}" name="y"
value="311"/>
            <mx:SetProperty target="{tlacitko}" name="label"
value="Zpět"/>
            <mx:SetProperty target="{obrazek}" name="x"
value="367"/>
            <mx:SetEventHandler target="{tlacitko}" name="click"
handler="currentState=''"/>
            <mx:AddChild relativeTo="{okno}"
position="lastChild">
                <mx:Text x="10" y="10" width="321" height="293"
id="text" fontSize="11">
                    <mx:text>text</mx:text>
                </mx:Text>
            </mx:AddChild>
            <mx:SetProperty target="{okno}" name="title"
value="Cumulonimbus - Cb (dešťová kupa)"/>
            <mx:SetProperty target="{obrazek}" name="width"
value="120"/>

```

```

        <mx:SetProperty target="{obrazek}" name="height"
value="91"/>
        <mx:SetStyle target="{okno}" name="fontSize"
value="20"/>
        <mx:SetStyle target="{tlacitko}" name="fontSize"
value="10"/>
        <mx:AddChild relativeTo="{okno}"
position="lastChild">
            <mx:Image x="367" y="109" width="120"
height="91" source="obr/cumulonimbus2.jpg" id="cirrus2"/>
        </mx:AddChild>
        <mx:AddChild relativeTo="{okno}"
position="lastChild">
            <mx:Image x="367" y="208" width="120"
height="91" source="obr/cumulonimbus3.JPG" id="cirrus3"/>
        </mx:AddChild>
        <mx:SetProperty name="height" value="440"/>
    </mx:State>
</mx:states>
<mx:transitions>
    <mx:Transition id="myTransition" fromState="*" toState="*">
        <mx:Parallel targets="{[okno,tlacitko,obrazek]}">
            <mx:Move duration="300"/>
            <mx:Resize duration="300"/>
        </mx:Parallel>
    </mx:Transition>
</mx:transitions>

    <mx:TitleWindow x="235" y="56" width="199" height="197"
layout="absolute" id="okno" title="Cumulonimbus" fontSize="20">
        <mx:Image x="36" y="10" width="107" height="84"
id="obrazek" source="obr/cumulonimbus1.JPG"/>
        <mx:Button x="42" y="102" label="Podrobnosti"
cornerRadius="12" color="0xffffffff" fillColors="[0x55C0FF,0x0050AA]"
fillAlphas="[1.0,1.0]" highlightAlphas="[1.0,0.2]" focusAlpha="0.2"
id="tlacitko" click="currentState='podrobnosti'" fontSize="10"/>
    </mx:TitleWindow>

</mx:Application>

```

Tvary oblaků – Accordion

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" width="650" height="600" cornerRadius="13"
horizontalAlign="center" verticalAlign="middle"
backgroundColor="#ADD1FF">

    <mx:Panel layout="absolute" color="#470094" width="630"
height="600" horizontalAlign="center" themeColor="#A200FF"
fontSize="20">

        <mx:Accordion id="accordion" color="#470094" width="100%"
height="530" x="0" y="0">

```



```

        <mx:VBox label="FIBRATUS - fib" fontSize="20"
color="#00060D" verticalAlign="middle">
        <mx:Text text width="578" height="54" fontSize="12"/>
        <mx:Image source="obr/fibratus1.jpg" width='150'
height='120' scaleContent="true"/>
        </mx:VBox>
        <mx:VBox label="UNCINUS - unc" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="69" fontSize="12"/>
        <mx:Image source="obr/uncinus1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="SPISSATUS - spi" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="70" fontSize="12"/>
        <mx:Image source="obr/spissatus1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="CASTELLANUS - cas" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="81" fontSize="12"/>
        <mx:Image source="obr/castellanus1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="FLOCCUS - flo" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="69" fontSize="12"/>
        <mx:Image source="obr/floccus1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="STRATIFORMIS - str" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="57" fontSize="12"/>
        <mx:Image source="obr/stratiformis1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="NEBULOSUS - neb" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="62" fontSize="12"/>
        <mx:Image source="obr/nebulosus1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="LENTICULARIS - len" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="81" fontSize="12"/>
        <mx:Image source="obr/lenticularis1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="FRACTUS - fra" fontSize="14"
color="#00060D" >
        <mx:Text text width="578" height="59" fontSize="12"/>
        <mx:Image source="obr/fractus1.jpg" width="150"
height="120"/>
        </mx:VBox>
        <mx:VBox label="HUMILIS - hum" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="59" fontSize="12"/>

```

```

        <mx:Image source="obr/humilis1.jpg" width="150"
height="120"/>
    </mx:VBox>
    <mx:VBox label="MEDIOCRIS - med" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="81" fontSize="12"/>
        <mx:Image source="obr/mediocris1.JPG" width="150"
height="120"/>
    </mx:VBox>
    <mx:VBox label="CONGESTUS - con" fontSize="14"
color="#00060D">
        <mx:Text text= width="578" height="67" fontSize="12"/>
        <mx:Image source="obr/congestus1.jpg" width="150"
height="120"/>
    </mx:VBox>
    <mx:VBox label="CALVUS - cal" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="65" fontSize="12"/>
        <mx:Image source="obr/calvus1.jpg" width="150"
height="120"/>
    </mx:VBox>
    <mx:VBox label="CAPILLATUS - cap" fontSize="14"
color="#00060D">
        <mx:Text text width="578" height="81" fontSize="12"/>
        <mx:Image source="obr/capillatus1.jpg" width="150"
height="120"/>
    </mx:VBox>
</mx:Accordion>

</mx:Panel>
</mx:Application>

```

Odkazy

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" width="650" height="1426" cornerRadius="13"
horizontalAlign="center" verticalAlign="middle"
backgroundColor="#ADD1FF">

    <mx:Script>
        <![CDATA[
            import flash.events.MouseEvent;
            public function doZoom(event:MouseEvent):void {
                if (zoomAll.isPlaying) {
                    zoomAll.reverse();
                }
                else {
                    zoomAll.play([event.target], event.type ==
MouseEvent.ROLL_OUT ? true : false);
                }
            }
            private function smoothImage(ev:Event):void{
                var bmp:Bitmap = ev.target.content as Bitmap;
                bmp.smoothing = true;
            }
        ]]>
    </mx:Script>

```

```

    }
  ]]>
</mx:Script>

<mx:Zoom id="zoomAll" zoomWidthTo="2" zoomHeightTo="2"
zoomWidthFrom=".5" zoomHeightFrom=".5"/>

<mx:Image id="flex" source="@Embed('obr/atlas_oblaku.gif')"
width="80" height="120" scaleX=".5" scaleY=".5"
rollOver="doZoom(event)" rollOut="doZoom(event)"
creationComplete="smoothImage(event);" x="305" y="234"/>
<mx:Text x="44.5" y="124" text="DVOŘÁK, Petr. Atlas oblaků. 1.
vyd. Cheb: Svět křídel, 2001. 128 s. ISBN 80-86808-42-0."
fontSize="12" fontStyle="normal"/>
<mx:Text x="272.5" y="37" text="Literatura" fontSize="20"
fontWeight="bold" textAlign="center"/>
<mx:Text x="57.5" y="407" text="BEDNÁŘ, Jan. Meteorologie. 1.
vyd. Praha: Portál, 2003. 224 s. ISBN 80-7178-653-5." fontSize="12"/>
<mx:Image x="309" y="505"
source="@Embed('obr/meteorologie.gif')" width="60" height="90"
scaleX=".5" scaleY=".5" rollOver="doZoom(event)"
rollOut="doZoom(event)" creationComplete="smoothImage(event);"/>
<mx:Text x="46.5" y="642" text="DVOŘÁK, Petr. Atlas počasí. 1.
vyd. Cheb: Svět křídel, 2003. 138 s. ISBN 80-86808-02-5."
fontSize="12"/>
<mx:Image x="309" y="751"
source="@Embed('obr/atlas_pocasi.gif')" width="80" height="120"
scaleX=".5" scaleY=".5" rollOver="doZoom(event)"
rollOut="doZoom(event)" creationComplete="smoothImage(event);"/>
<mx:Text x="18.5" y="909" text="SVĚTOVÁ METEOROLOGICKÁ
ORGANIZACE. Mezinárodní atlas oblaků. Hydrometeorologický ústav"
fontSize="12"/>
<mx:Image x="315.65" y="1015"
source="@Embed('obr/atlas_mezinarodni.JPG')" width="120" height="80"
scaleX=".5" scaleY=".5" rollOver="doZoom(event)"
rollOut="doZoom(event)" creationComplete="smoothImage(event);"/>
<mx:Text x="25.5" y="1191" text="ROTH, G., D.. Encyklopedie
počasí. 1. vyd. Praha: Knižní klub, 2000. 296 s. ISBN 80-242-0228-X."
fontSize="12"/>
<mx:Image x="309" y="1304"
source="@Embed('obr/encyklopedie_pocasi.gif')" width="60" height="90"
scaleX=".5" scaleY=".5" rollOver="doZoom(event)"
rollOut="doZoom(event)" creationComplete="smoothImage(event);"/>
</mx:Application>

```